ETSI TS 103 999-2 V15.0.0 (2021-12)

Smart Secure Platform (SSP);
Part 2: Integrated SSP (iSSP) characteristics Test Specification
Release 15

Reference
DTS/SCP-0000TSSPvf00-2

Keywords
iSSP, 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

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.

The present document is part 2 of a multi-part deliverable covering testing aspects for the Smart Secure Platform (SSP), as identified below:

Part 1: "Test Specification, general characteristics";

Part 2: "Integrated SSP (iSSP) characteristics Test Specification".

Modal verbs terminology

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

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

1 Scope

The present document details the test specifications for the Smart Secure Platform (SSP) integrated into an SoC, also known as iSSP. It specifies the test environment to verify conformance requirements for services running in the Smart Secure Platform and in any terminal hosting a Smart Secure Platform application as defined in ETSI TS 103 666-1 [9] focusing on the specific attributes that are defined for the iSSP in ETSI TS 103 666-2 [10].

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] ANSI X9.62-2005: "Public Key Cryptography for the Financial Services Industry, The Elliptic Curve Digital Signature Algorithm (ECDSA)".

[2] BSI-CC-PP-0084-2014: "Security IC Platform Protection Profile with Augmentation Packages".

[3] BSI TR-03111: "Elliptic Curve Cryptography", Version 2.10.

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

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

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

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

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

[9] ETSI TS 103 666-1: "Smart Secure Platform (SSP); Part 1: General characteristics".

[10] ETSI TS 103 666-2: "Smart Secure Platform (SSP); Part 2: Integrated SSP (iSSP) characteristics".

[11] ETSI TS 103 999-1: "Smart Secure Platform (SSP); Part 1: Test Specification, general characteristics".

[12] GlobalPlatform Technology: "Card Specification", Version 2.3.1.

[13] GlobalPlatform Technology: "Open Firmware Loader for Tamper Resistant Element", Version 1.3.

[14] GlobalPlatform Technology: "Virtual Primary Platform - Firmware Format", Version 1.0.1.

[15] GlobalPlatform Technology: "VPP - Concepts and Interfaces", Version 1.0.1.

[16] GlobalPlatform Technology: "VPP - OFL VNP Extension", Version 1.0.

[17] IETF draft-shen-sm2-ecdsa-02: "SM2 Digital Signature Algorithm".

[18] IETF RFC 3986: "Uniform Resource Identifier (URI): Generic Syntax".

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

[20] IETF RFC 4868: "Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec".

[21] IETF RFC 5280: "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile".

[22] IETF RFC 5480: "Elliptic Curve Cryptography Subject Public Key Information".

[23] IETF RFC 5639: "Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation".

[24] IETF RFC 5754: "Using SHA2 Algorithms with Cryptographic Message Syntax".

[25] IETF RFC 5758: "Internet X.509 Public Key Infrastructure: Additional Algorithms and Identifiers for DSA and ECDSA".

[26] IETF RFC 7540: "Hypertext Transfer Protocol Version 2 (HTTP/2)".

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

[28] ISO/IEC 10118-3:2018: "IT Security techniques - Hash-functions - Part 3: Dedicated hash functions".

[29] ISO/IEC 14888-3:2018: "IT Security techniques - Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms".

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

[31] NIST 800-56A (May 2013): "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography (Revision 2)".

[32] NIST 800-108: "Recommendation for Key Derivation Using Pseudorandom Functions".

[33] NIST SP 800-38B (May 2005): "Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication".

[34] ETSI SCP forge repository - ETSI TS 103 999-2 Projects:.

- ETSI TS 103 999-2 iSSP Test Specification; available at: https://forge.etsi.org/rep/scp/ts_103999-2_issp_testspec
- ETSI TS 103 999-2 iSSP Test Tool; available at: https://forge.etsi.org/rep/scp/ts_103999-2_issp_Testtool
- ETSI TS 103 999-2 iSSP eGCM; available at: https://forge.etsi.org/rep/scp/ts_103999-2_iSSP_eGCM

2.2 Informative references

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

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

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

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

Not applicable.

3 Definition of terms, symbols, abbreviations and formats

3.1 Terms

For the purposes of the present document, the terms given in ETSI TS 103 465 [8], ETSI TS 103 666-1 [9] and ETSI TS 103 666-2 [10] apply.

3.2 Symbols

For the purposes of the present document, the symbols given in ETSI TS 103 465 [8], ETSI TS 103 666-1 [9] and ETSI TS 103 666-2 [10] apply.

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in ETSI TS 103 465 [8], ETSI TS 103 666-1 [9], ETSI TS 103 666-2 [10] and the following apply:

ARP	Access Right Pattern
CI	Certificate Issuer
LBA	Local Bundle Assistant
SCL	SSP Common Layer
SKID	Subject Key IDentifier
SPBL	Secondary Platform Bundle Loader
SPBM	Secondary Platform Bundle Manager
SSP	Smart Secure Platform
SUT	System Under Test
TT	Test Tool

3.4 Formats

3.4.1 Format of the conformance requirement tables

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

Table 3.1: Format of the conformance requirement tables

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 requirement terms: RQ<XX><YY>_<ZZZ> or RQ<XX><YY>_<ZZZA> XX: Main clause of the core specification in which the conformance requirement is listed. YY: Subclause of the main clause in the core specification in which the conformance requirement is listed. ZZZ: Continuously increasing number starting with '001'. ZZZA: Sub-numbering (alphabetic) used if an identified requirement is split for clarification.
Clause	The "Clause" column helps to identify the location of a requirement by listing the clause hierarchy down to the subclause the requirement is located in.
Description	In this column the requirement text is shown. Where the text can either be a copy of the original requirement as found in ETSI TS 103 666-2 [10] or a text analogous to the requirement text (e.g.: if the requirement text is descriptive and can be shortened or truncated).

3.4.2 Format of the applicability table

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

Table 3.2: Format of the applicability table

Column	Meaning
Test ID	A reference to the test description identification 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. In general, the description text used will equal the test description name used in the present document.
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 [30]: 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 pro forma 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.

3.4.3 Numbers and Strings

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

Table 3.3: 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.	

3.4.4 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

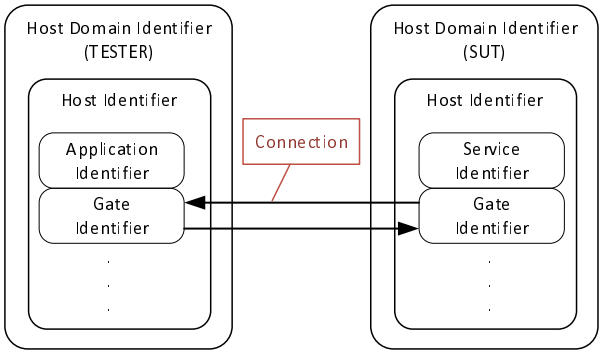
X.Y.1 Configurations

This header is to be 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>_<y>1<n> <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, an <y> for the clause number, a '1' for the 'Configurations' clause, and <n>, a minimum one-digit configuration number. 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>_<y>1<n>
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 to be 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>_<y>2<n> <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, an <y> for the clause number, a '2' for the 'Procedures' clause, and <n>, a minimum one-digit configuration number. 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>_<y>2<n>
Procedure objectives	Description of the procedure objectives.
Configuration reference	C<aaa>_<y>1<n> 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 note2.	
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 to be used for every clause that includes test descriptions. It may be followed by:

X.Y.3.1 <aaa>_<y>3<optional s><n> <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, an <y> for the clause number, a '3' for the 'Test descriptions' clause, a clause number <s> (optional – only added if test descriptions are structures in sub-subclauses) and <n>, a one-digit configuration number. 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>_<y>3<s><n> or <aaa>_<y>3<n>
Test objectives	Description of the test objectives. See note 1.
Configuration reference	C<aaa>_<y>2<n> 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 should 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 shall be defined by existing procedures. Referenced procedures are intended to be executed in given order.		

Requirement IDs listed in the Req.ID tab 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 or if the described test step could be executed successfully. 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 clause shall inform about the status of the requirements. E.g.:

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

The header of this clause shall be adjusted depending on which condition applies for the identified requirements.

Example text for requirements referenced from another standardization body:

The following requirements identified in <XYZ> are not tested in accordance with the present document, as they are referencing requirements from another 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 clause in the Test description clause. Nevertheless, it can be provided as the first 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 may be done on any level of the test description hierarchy. Furthermore, it is not required to generate sub-groups for all the three main sections (Configurations, Procedures, Test descriptions) if adding a sub-group is useful in any of these sections.

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.

3.4.5 Dynamic content validation in ASN.1 structure

In certain test descriptions dynamic content returned by the DUT (e.g.: value within ASN.1 structure, signature, integer, …) is processed according to the following 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 identifying the operation to be performed on dynamic content of aFieldName as:
  - STORE: store the dynamic content of an aFieldName into a test tool variable identified by the Variable_identifier.
  - REPLACE: retrieve a variable identified by the Variable_identifier and replace the content of aFieldName with the content of the variable.
  - COMPARE: compare the content of aFieldName with the content of a variable and return 'true' or 'false' as a result to the test tool. This operator requires one or more parameters. If more than one parameter is used, the parameters are OR concatenated.

    Possible parameters are:
    - 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 of a set of alphanumeric characters only.

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

aFieldName … / operations */*

For example:

```
aHandleNotificationHeader {
    aNotificationReceiverId eFUNCTION-REQUESTER-ID-1,
    aNotificationCallId '00000000'H /* <COMPARE(aEMPTY,DIF)>*/,
  },
```

where:

```
aEMPTY OCTET STRING ::= ''H /*<STORE(aEMPTY)>*/
```

4 Test environments

4.1 Test environments for the different test aspects

4.1.0 General overview on the iSSP ecosystem to be tested

The general architecture of the iSSP ecosystem is defined in ETSI TS 103 666-1 [9], clause 12.1.

A representation of the iSSP ecosystem is shown in Figure 4.1 to ease the identification of entities required and interfaces tested in accordance with the present document. Interfaces (Si1, Si2, Si3 and Si4) involved in Secondary Platform Bundle management are highlighted.

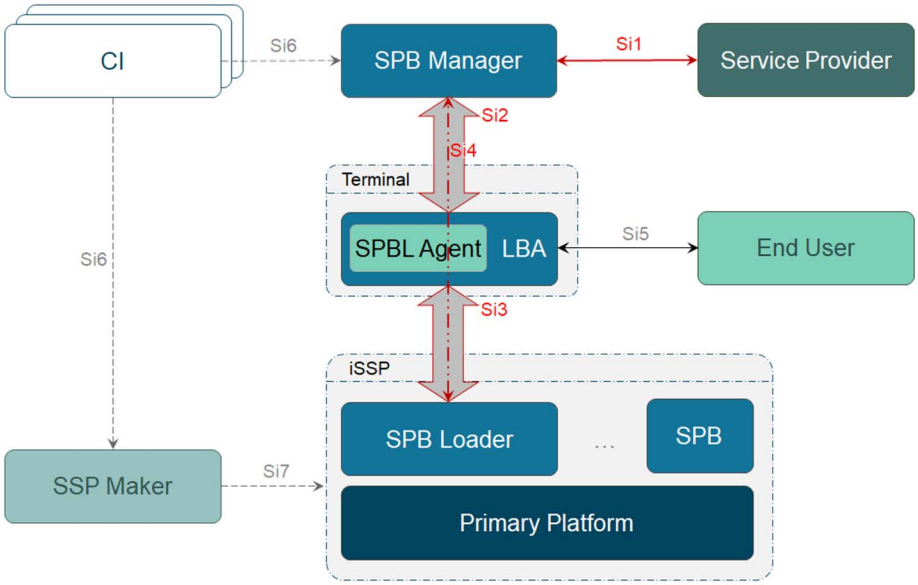


Figure 4.1: iSSP ecosystem

4.1.1 Evaluation Assurance Level certification

The support of a certification by composition from the SSP Primary Platform Evaluation Assurance Level is defined in ETSI TS 103 666-1 [9], clause 11.2.1.

SSP Evaluation Assurance Level certification is out of scope of the present document.

4.1.2 Test environment for Secondary Platform Bundle services

The test environment defined in Figure 4.2 illustrates the perspective of the tests of a service running in the SSP from an application running on the terminal.

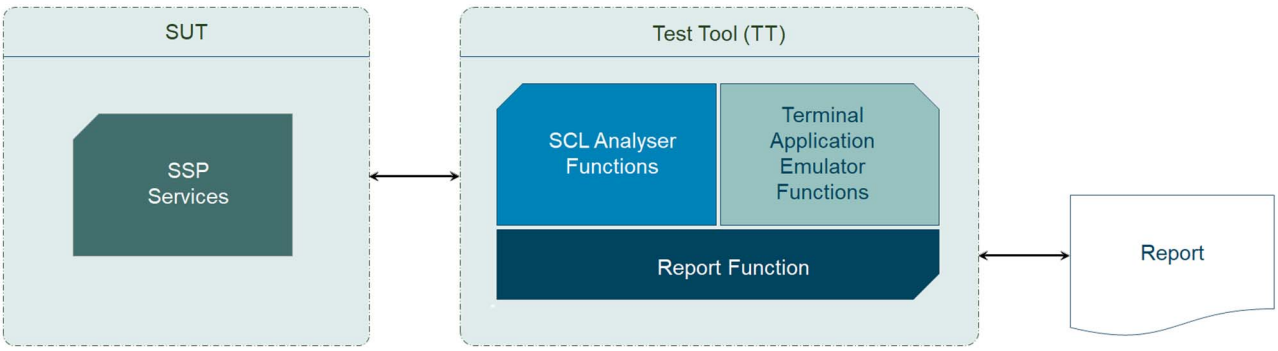


Figure 4.2: Tests of a service in the SSP

All tests defined in ETSI TS 103 999-1 [11] are applicable.

This test environment is valid for testing the SPB loader service described in ETSI TS 103 999-1 [11], clause 12 and will support the Si3 interface.

NOTE: The test environment defined for testing a service in the SSP in the present document is similar to the one defined for testing a service in the SSP in ETSI TS 103 999-1 [11], clause 4, Figure 4.3.

4.1.3 Test environment for Secondary Platform Bundle Manager services

The test environment defined in Figure 4.3 illustrates the perspective of the tests of services running in the SPBM (SUT).

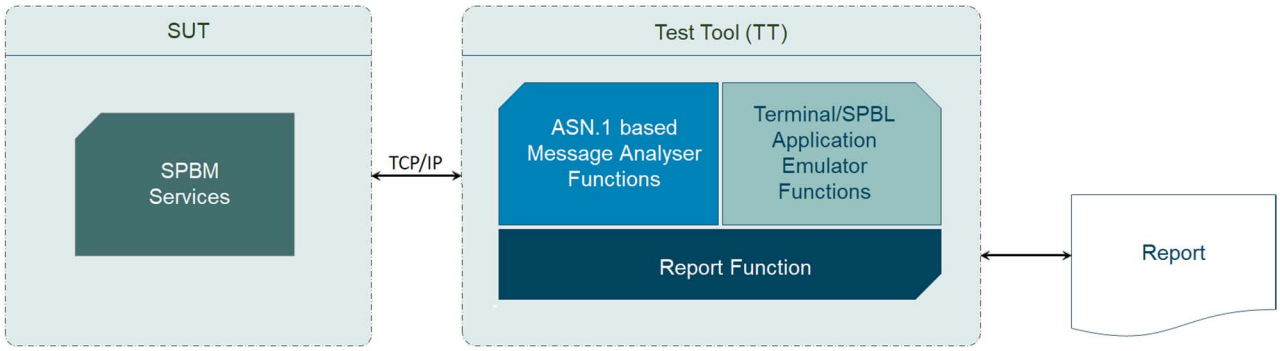


Figure 4.3: Tests of a service in the SPBM

The applications running in the test tool are functionally equivalent to:

- The LBA running on the terminal via Si2 and Si4.
- The service provider via the Si1.

As the list of functions table 12.4 in ETSI TS 103 666-2 [10] does not define any service in the LBA, no test descriptions for testing the LBA are needed.

The test tool connector is the Si1 and Si2 interfaces as defined in ETSI TS 103 666-2 [10], clauses 12.6.3 and 12.6.4.

The SPBM shall be prepared for test purposes in supporting a set of certificates for ETSI tests. These certificates shall be compliant to what is defined in ETSI TS 103 666-2 [10], clause 12.2.1.

The SPBM is a certified functional block for which no invasive test tool connector is allowed. Consequently, the Si4 interface functionality is tested with negative cases, deducing the transfer of protocol elements required for authorization, mutual authentication, integrity and confidentiality.

The testing of the Si6 interface connecting the CI and the SPBM is out of the scope of the present document.

4.1.4 Test environment for Primary Platform services

Figure 4.4.4 illustrates the perspective of the tests of a service running in the primary platform from an application running on the primary platform point of view (here a SPB).

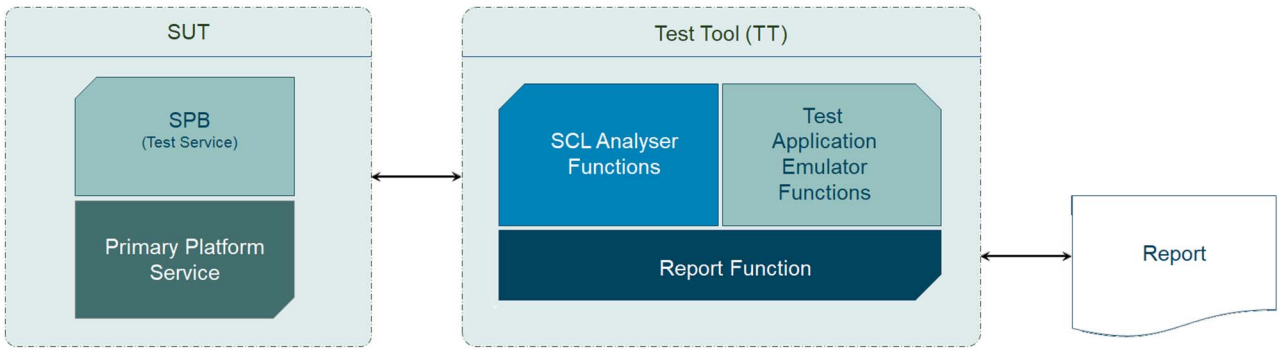


Figure 4.4: Tests of a service in the primary platform

The primary platform can only be tested from a SPB via the interface defined in ETSI TS 103 666-2 [10], clause 8. The iSSP shall enable the SPB and then be capable to address test content to the Primary Platform. The SPB Test Service interprets commands from the test application running in the test tool.

Tests related to the kernel functions of the ABI/API and to the communication service interface of the Primary Platform are out of the scope of the present document.

4.1.5 Principles of the data exchange

4.1.5.1 Data Format verification

The verification of the data exchanged between the SPBL, the SPBM and the LBA is globally performed by comparing the data in question with the ASN.1 model defined in ETSI TS 103 666-2 [10]. The application acts as data issuer when the connected service is the data receiver and vice versa. The data flow from the data issuer to the data receiver is illustrated in Figure 4.5. The data format verification throws an exception if the exchanged data are not fully compliant with the ASN.1 model.

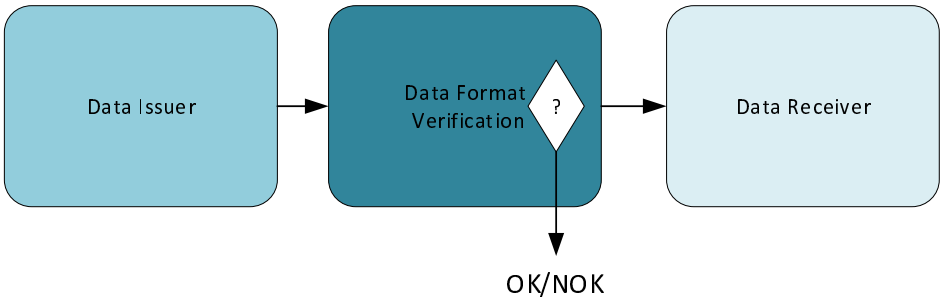


Figure 4.5: Data format verification

The data format verification on the presentation layer of the Si2 interface as shown in Figure 4.5 is done for ASN.1 compliance only. The correctness of the data contents conveyed by the Si2 presentation layer are verified by the TT by analysing the received data.

4.1.5.2 Data contents verification

4.1.5.2.1 SUT test concept

The TT always acts as the application. As the SUT (service) appears as a "black box", checking its functionalities is done by stimulating the SUT with invalid data contents provided in an appropriate ASN.1 model, expecting the SUT to throw errors and/or exceptions.

To verify that the SUT is doing data contents verification software tools allowing to provide invalid or incorrect data using the appropriate ASN.1 model are provided.

4.1.5.2.2 Software tools for clause 12

Example software tools associated with the Si2 test descriptions provided in the present document are available in the ETSI forge repository [34]. All PDU use the DER format.

The provided software tools enable a TT to generate:

- The SPBL certification path (authentic) leading to a correct certification path.
- The SPBM certification path (fake) leading to a wrong certification path.
- The SPBL certification path (authentic) leading to a correct certification path.
- The SPBM certification path (fake) leading to a wrong certification path.
- The Si2GetSpbmCertificate command according to parameters.
- The Si2GetSpbmCertificate response according to parameters.
- The Si2GetBoundSpbImage command according to parameters.

- The Si2GetBoundSpbImage response according to parameters.
- The Si2HandleNotification command according to parameters.
- The Si2HandleNotification response according to parameters.

The provided software tools enable a TT to verify:

- The Si2GetSpbmCertificate response according to parameters.
- The Si2GetBoundSpbImage response according to parameters.
- The Si2HandleNotification response according to parameters.

Software tools associated with these test descriptions do not deal with the firmware as defined by GlobalPlatform in Virtual Primary Platform - Firmware Format, [14].

4.1.5.3 Public Key Infrastructure for tests

Figure 4.6 defines the PKI used for the test descriptions of the Si4 interface.

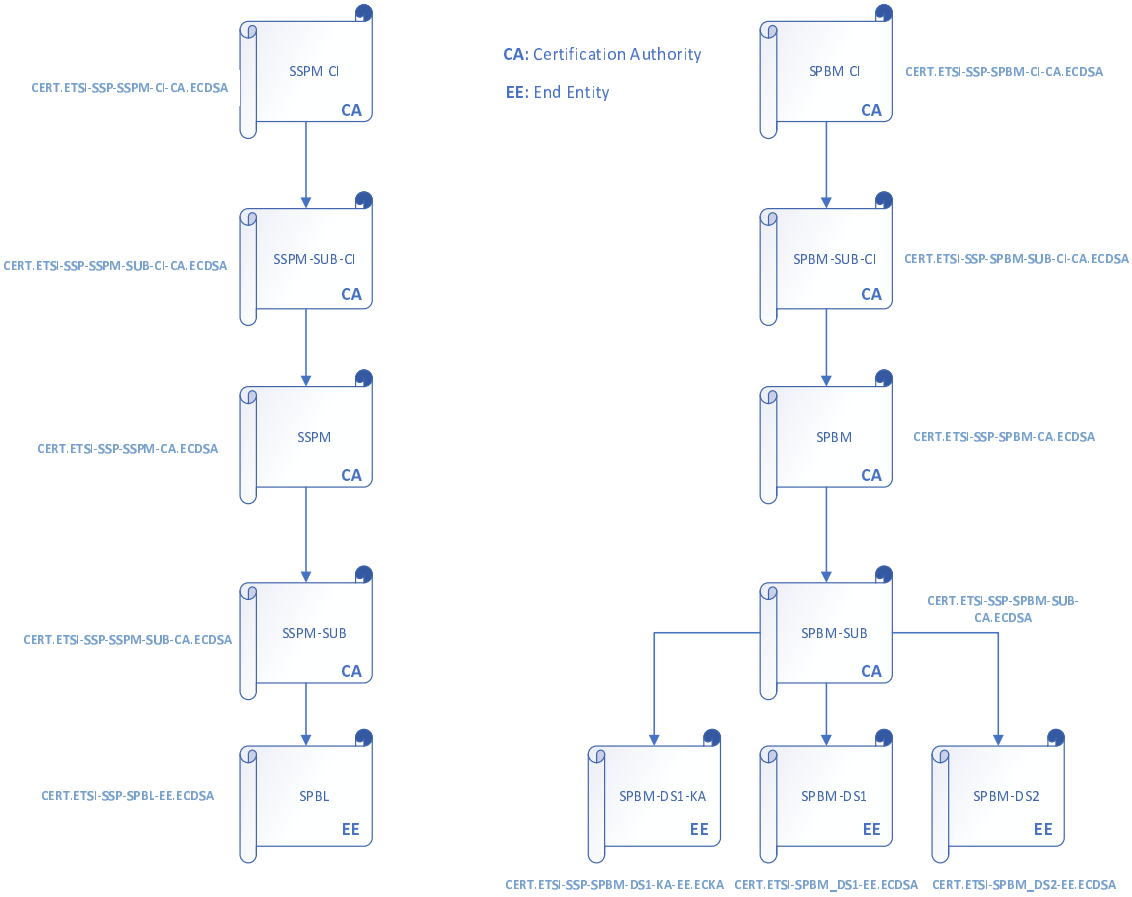


Figure 4.6: PKI for tests

The hierarchical list of the digital certificates in the certification path of the SSPM CI is the following:

- CERT.ETSI-SSP-SSPM-CI-CA.ECDSA
- CERT.ETSI-SSP-SSPM-SUB-CI-CA.ECDSA
- CERT.ETSI-SSP-SSPM-CA.ECDSA

- CERT.ETSI-SSP-SSPM-SUB-CA.ECDSA
- CERT.ETSI-SSP-SPBL-EE.ECDSA

The hierarchical list of the digital certificates in the certification path of the SPBM CI is the following:

- CERT.ETSI-SSP-SPBM-CI-CA.ECDSA
- CERT.ETSI-SSP-SPBM-SUB-CI-CA.ECDSA
- CERT.ETSI-SSP-SPBM-CA.ECDSA
- CERT.ETSI-SSP-SPBM-SUB-CA.ECDSA
- CERT.ETSI-SSP-DS1-EE.ECDSA, CERT.ETSI-SSP-DS2-EE.ECDSA, CERT.ETSI-SSP-DS1-KA-EE.ECKA

For tests purposes only, a set of private keys compliant with the public key lengths supported by the ETSI TS 103 666-2 [10] is available in the ETSI forge repository as defined in SCP iSSP tooling [34]. The Si4 security protocol is independent of the ECC key lengths as well the Si4 test descriptions.

4.1.6 Common ASN.1 coding

```
ETSITestGlobalDefinitions { id-issp test(3) }
DEFINITIONS
AUTOMATIC TAGS
EXTENSIBILITY IMPLIED ::=
BEGIN
EXPORTS ALL;
IMPORTS
UUID,
id-issp
FROM ISSPDefinitions;
/* Imports */
id-issp-test                OBJECT IDENTIFIER ::= {id-issp test(3)}
issp-egcm                   OBJECT IDENTIFIER ::= {id-issp-test egcm (1)}
issp-egcm-aes-128           OBJECT IDENTIFIER ::= {issp-egcm egcm-aes-128 (1)}
issp-egcm-aes-256           OBJECT IDENTIFIER ::= {issp-egcm egcm-aes-256 (2)}
/*Custodian for tests*/
issp-acustodian-oid         OBJECT IDENTIFIER ::= {id-issp-test acustodian-oid (2)}
issp-acustodian-oid-telecom OBJECT IDENTIFIER ::= {issp-acustodian-oid telecom (1)}

id-globalplatform           OBJECT IDENTIFIER ::= {iso(1) member-body(2) us(840)
globalplatform(114283)}
id-part-number              OBJECT IDENTIFIER ::= {id-globalplatform ofl(10) pn(1)}
/*FamilyId Oid*/
id-family-id-test           OBJECT IDENTIFIER ::= {id-issp-test id-family-id(3)}
id-family-id-test-1         OBJECT IDENTIFIER ::= {id-family-id-test  family(1)}
id-family-id-test-2         OBJECT IDENTIFIER ::= {id-family-id-test  family(2)}

/* Family UUIDv5 for tests*/
/*URN: urn:ttf001.etsi.org:id-family-id-test-1*/
eFamilyIdTest1 UUID ::= 'FF334CCD9D055649B517C6ECBB1B5383'H
/*URN: urn:ttf001.etsi.org:id-family-id-test-2*/
eFamilyIdTest2 UUID ::= '58DBFEA315355BBAA732D43F6C0A2956'H

/* SPBId UUIDv5 for tests*/
/*URN: urn:ttf001.etsi.org:codem:47929dd4-9854-4f71-8dd8-e247fd909e13 */
eSPBIdTest1 UUID ::= 'E044EB70B41359DD9399F3D4124555E0'H
/*URN: urn:ttf001.etsi.org:codem:83e58fe0-35ea-47f6-9b74-bdb7a5ecb772 */
eSPBIdTest2 UUID ::= '1266BF3477E251BEB02D23C7478B5AAD'H

END
```

ASN.1 coding/SCP iSSP tooling can be found in a sub-folder of [34] at: https://forge.etsi.org/rep/scp/ts_103999-2_issp_testspec/raw/master/TS103999-2.asn.

NOTE: Opening to the referenced file might only work if entered into the address bar of your internet browser.

4.2 Applicability Table

The applicability tables in this clause are formatted as described in clause 3.4.2.

Table 4.1: Applicability table

Test ID	Description	Release	Rel-15	Support
PSVC_322	Primary Platform - Open a pipe session on the SPBL Service Gate	Rel-15	M	
SVC_3311 - SVC_33110	Secondary Platform Bundle Loader	Rel-15	M	
iSP_4341 - iSP_4344	Secondary Platform - Capability exchange	Rel-15	M	
SI1_63311 - SI1_63317	Si1 interface - Si1.CreateSPReference	Rel-15	M	
SI1_63321 - SI1_63328	Si1 interface - Si1.SelectSpb	Rel-15	M	
SI1_63331 - SI1_63333	Si1 interface - Si1.FinalizePreparation	Rel-15	M	
SI1_63341 - SI1_63347	Si1 interface - Si1.CancelPreparation	Rel-15	M	
SI1_63351	Si1 interface - Si1.HandleNotification	Rel-15	M	
SI2_64311 - SI2_643110	Si2 interface - Si2.GetSpbmCertificate	Rel-15	M	
SI2_64321 - SI2_64327	Si2 interface - Si2.GetBoundSpbImage	Rel-15	M	
SI2_64331	Si2 interface - Si2.HandleNotificatio	Rel-15	M	
SI3_65311 - SI3_65314	Si3 interface - Si3.GetSspInfo	Rel-15	M	
SI3_65321	Si3 interface - Si3.SetSpbmCredential	Rel-15	M	
SI3_65331	Si3 interface - Si3.LoadBoundSpbInfo	Rel-15	M	
SI3_65341	Si3 interface - Si3.LoadBoundSpbSds	Rel-15	M	
SI3_65351	Si3 interface - Si3.LoadBoundSpbSeg	Rel-15	M	
SI3_65361	Si3 interface - Si3.GetSspCredential	Rel-15	M	
SI3_65371 - SI3_65372	Si3 interface - Si3.EnableSpb	Rel-15	M	
SI3_65381	Si3 interface - Si3.DisableSpb	Rel-15	M	
SI3_65391	Si3 interface - Si3.DeleteSpb	Rel-15	M	
SI3_653101	Si3 interface - Si3.GetSpbMetadata	Rel-15	M	
SI3_653111	Si3 interface - Si3.UpdateSpbState	Rel-15	M	
SI3_653121	Si3 interface - Si3.GetSpbState	Rel-15	M	
SI3_653131	Si3 interface - Si3.SwitchSpb	Rel-15	M	
SI3_653141	Si3 interface - SPB Management Operations	Rel-15	M	
SI3_65321	Si3 interface - Si3.SetSpbmCredential	Rel-15	M	
SI4_66311	Si4 interface - Si4.SPBL service	Rel-15	M	
SI4_66321	Si4 interface - Si4.SPB Manager service	Rel-15	M	

5 Conformance requirements

5.1 Conformance requirement references

The conformance requirements that apply to the test descriptions defined in the present document are derived from the specification named in the reference text preceding each conformance requirement listing.

5.2 Juxtaposition of terminologies

ETSI TS 103 666-2 [10] is using a different terminology than the Open Firmware Loader (OFL) specification [13] from Global Platform. As the Global Platform specification is referenced for various commands and functions, the juxtaposition of the used terms shall help to understand the test descriptions defined within the present document.

Table 4.2: Juxtaposition of ETSI and Global Platform terms

ETSI	OLF
SSP Maker	TRE maker
SSP	TRE
SPBL	OFL
SPB container	Firmware
SPL certificate	OFL certificate (CERT.OFL.ECDSA)
SPBM (SPB Manager)	IDS (Image Delivery Server)
SPBM KA certificate	CERT.IDS1.ECKA
Primary Platform identifier	No equivalence in OFL
Si1.SelectSpb	Out of the scope of GlobalPlatform
Si1.CreateSPReference	Out of the scope of GlobalPlatform
Si3.GetSspInfo	ANY_GET_PARAMETER with parameters for reading the registry
Si2.GetSpbmCertificate	Out of the scope of GlobalPlatform
Si3.SetSpbmCredential	ANY_SET_PARAMETER with parameter for IDS_CREDENTIALS
Si2.GetBoundSpbImage	Out of the scope of GlobalPlatform
aSspInfoProtected	ANY_GET_PARAMETER with TRE_CREDENTIALS
aBoundSpbImageByTransacId	Out of the scope of GlobalPlatform
Si3.LoadBoundSpbInfo	OFL_DO_OPERATE(VNP)
Si3.LoadBoundSpbSds	OFL_CHANGE_SEGMENT(VNP)
Si3.LoadBoundSpbSeg	OFL_LOAD_SEGMENT(VNP)
aChangeSegmentParameter	SDS (Segment Descriptor Structure)
aDoOperateParameter	IMD (Image Descriptor)
aLoadSegmentParameter	FFS
LBA	OFL Agent
bound Secondary Platform Bundle image	Bound Image
Si3.EnableSpb	OFL_ENABLE_FIRMWARE (VNP)
Si3.DisableSpb	OFL_DISABLE_FIRMWARE
Si3.DeleteSpb	OFL_DELETE_SESSION
Si3.GetSpbMetadata	ANY_GET_PARAMETER with register
SPB_STATE	ANY_GET_PARAMETER with the OFL_STATE register
Si2.HandleNotification	Out of scope of OFL
aPartNumberId	ANY_GET_PARAMETER with the PART_NUMBER register
aPpIdentifier	No match in OFL
aFamilySpecificSspInfo	No match in OFL
Si3.GetSspCredential	ANY_GET_PARAMETER with the TRE_CREDENTIAL_PARAMETER register
aChallengeS	CHALLENGE_S
aIdTransac	ID_TRANSAC
aEPkSpblKa	PK.OFL.ECKA
aM-SSP	M1, H1
aImageOwnerId	IMOL
aNumberSegment	NUM_SEG in IMD
aEncryptionType	In the ATK.IDS2.ECDHE
aImageMakerId	UUIDI
aM-IMD	M2, H2
aM-ARP	M3, H3
aM-TimeStamp	M4, H4
aSpbmToken	ATK.IDS2.ECDHE

5.3 Overview - Security requirements

Reference: ETSI TS 103 666-2 [10], clause 5.2.

Req.ID	Clause	Description
RQ0502_001	5.2	The provisions of ETSI TS 103 666-1 [9], clause 6.11 shall apply.
RQ0502_002	5.2	The software and sensitive data of the iSSP shall never be exposed from the iSSP to any external component in plain text.
RQ0502_003	5.2	The protection of software and sensitive data shall provide privacy, confidentiality, integrity, protection against rollback attacks, and protection against side-channel attacks.
RQ0502_004	5.2	In the case where software and sensitive data are stored outside the iSSP, they shall also be protected in a way to achieve perfect forward secrecy and they shall be securely bound to that given iSSP instance, in accordance to clause 7.1.3.4 of ETSI TS 103 666-2 [10].

5.4 iSSP Architecture

Reference: ETSI TS 103 666-2 [10], clause 6.

Req.ID	Clause	Description
	6.1	***Overview***
RQ0601_001	6.1	The provisions of GlobalPlatform VPP - Concepts and Interfaces [15], clause 2.1 shall apply.
	6.2	***Functional architecture***
RQ0602_001	6.2	The provisions of GlobalPlatform VPP - Concepts and Interfaces [15], clauses 5.1 and 5.2 shall apply.
	6.3	***Security perimeters***
RQ0603_001	6.3	The provisions of GlobalPlatform VPP - Concepts and Interfaces [15], clause 5.3 shall apply.
	6.4	***Unprivileged execution model***
RQ0604_001	6.4	The provisions of GlobalPlatform VPP - Concepts and Interfaces [15], clause 5.4 shall apply.
	6.5	***Unprivileged virtual address space***
RQ0605_001	6.5	The provisions of GlobalPlatform VPP - Concepts and Interfaces [15], clause 5.5 shall apply.
	6.6	***Run time model***
RQ0606_001	6.6	The provisions of GlobalPlatform VPP - Concepts and Interfaces [15], clause 5.6 shall apply.

5.5 Primary Platform

5.5.1 Hardware Platform

Reference: ETSI TS 103 666-2 [10], clause 7.1.

Req.ID	Clause	Description
	7.1.1	***Architecture***
RQ0701_001	7.1.1	The provisions of GlobalPlatform VPP - Concepts and Interfaces [15], clause 3.1 shall apply with the exception that the presence of the SoC shown in figure 3-1 of [15] is mandatory.
RQ0701_002	7.1.1	The provisions of GlobalPlatform VPP - Concepts and Interfaces [15], clause 3.1 shall apply with the exception that the iSSP shall contain an autonomous and independent clock system.
RQ0701_003	7.1.1	The provisions of GlobalPlatform VPP - Concepts and Interfaces [15], clause 3.1 shall apply with the exception that the iSSP shall contain communication functions.
RQ0701_004	7.1.1	The provisions of GlobalPlatform VPP - Concepts and Interfaces [15], clause 3.1 shall apply with the exception that the iSSP may contain the data protection hardware function.

Req.ID	Clause	Description
	7.1.3	***Security functions***
RQ0701_005	7.1.3.1	The provisions of GlobalPlatform VPP - Concepts and Interfaces [15], clause 3.4.1 SREQ19 shall apply.
RQ0701_006	7.1.3.2	The Primary Platform shall provide a Memory Management Function (MMF) to avoid dependency of the Secondary Platform Bundle design with respect to the execution memory addressing.
RQ0701_007	7.1.3.2	The provisions of GlobalPlatform VPP - Concepts and Interfaces [15], clauses 3.2.2 and 3.5 shall apply.
RQ0701_008	7.1.3.3	The hardware platform shall provide a hardware function for protecting its long term keys as defined in GlobalPlatform VPP - Concepts and Interfaces [15], clause 3.4.3.
RQ0701_009	7.1.3.3	The key protection function shall perform key derivation, as specified in NIST 800-108 [32], with robustness of the PRF equivalent to or greater than HMAC-SHA-256 as defined in IETF RFC 4868 [20] or CMAC as defined in NIST SP 800-38B [33].
RQ0701_010	7.1.3.3	The long-term seed value shall be accessible only by the hardware platform. The probability that two distinct hardware platforms have the same long term seed shall be negligible.
RQ0701_011	7.1.3.3	The hardware platform shall provide a data path for the key protection function output.
RQ0701_012	7.1.3.3	The key protection function output shall be made available for the data protection hardware function described in ETSI TS 103 666-2 [10] clause 7.1.3.4, if that clause is supported, or to the cryptographic functions described in ETSI TS 103 666-2 [10] clause 7.1.7.
RQ0701_013	7.1.3.4	The support of a hardware function performing the encryption to export software and data outside the iSSP shall only be accessible by the low-level Operating System in the Primary Platform. If this hardware function is supported, the provisions of GlobalPlatform VPP - Concepts and Interfaces [15], clause 3.4.2 shall apply.
RQ0701_014	7.1.3.4	For the purpose of storing and verifying software and data outside the iSSP only keys provided by the key protection function shall be used. If this hardware function is supported, the provisions of GlobalPlatform VPP - Concepts and Interfaces [15], clause 3.4.2 shall apply.
RQ0701_015	7.1.3.5	The provisions of GlobalPlatform VPP - Concepts and Interfaces [15], clause 3.2.6 shall apply.
RQ0701_016	7.1.3.6	The hardware platform shall protect against the disclosure of keys managed by the Primary Platform, when using test functions of the SoC or test equipment.
RQ0701_017	7.1.3.5	The provisions of GlobalPlatform VPP - Concepts and Interfaces [15], clause 3.7 shall apply.
RQ0701_018	7.1.3.8	The provisions of GlobalPlatform VPP - Concepts and Interfaces [15], clause 3.4.4 shall apply.
	7.1.4	***Memories***
RQ0701_019	7.1.4.1	The provisions of GlobalPlatform VPP - Concepts and Interfaces [15], clause 3.2.3 shall apply.
RQ0701_020	7.1.4.1	The Primary Platform shall provide the Secondary Platform with direct memory-mapped access to the NVM, whether the NVM is integrated in the iSSP (iNVM) or accessed remotely (rNVM).
RQ0701_021	7.1.4.2	The Primary Platform shall provide the Secondary Platform with direct memory-mapped access to the volatile memory, whether the memory is integrated in the iSSP (iRAM) or accessed remotely (rRAM).
	7.1.7	***Cryptographic functions***
RQ0701_022	7.1.7	The provisions of GlobalPlatform VPP - Concepts and Interfaces [15], clause 3.2.7 shall apply.
	7.1.8	***Clock***
RQ0701_023	7.1.8	The iSSP shall embed an autonomous and independent clock system in conformance with the Protection Profile BSI-CC-PP-0084-2014 [2].
RQ0701_024	7.1.8	The provisions of ETSI TS 103 666-1 [9], clause 6.3 shall apply.
	7.1.9	***SSP internal interconnect***
RQ0701_025	7.1.9	All elements contained in the iSSP shall only be physically connected to other elements in the iSSP, except as specified in clause 7.1.5 of ETSI TS 103 666-1 [9].
	7.1.10	***Secure CPU***
RQ0701_026	7.1.10	The hardware platform shall contain one or more dedicated CPUs, which are inside the iSSP and separated from the rest of the SoC.
RQ0701_027	7.1.10	The provisions of GlobalPlatform VPP - Concepts and Interfaces [15], clause 3.2.1 shall apply.
RQ0701_028	7.1.10	The CPU(s) shall be based at least on a 32-bit architecture.

Req.ID	Clause	Description
	7.1.11	***Random Number Generator***
RQ0701_029	7.1.11	The provisions of GlobalPlatform VPP - Concepts and Interfaces [15], clause 3.2.8 shall apply.

5.5.2 Low-level Operating System

Reference: ETSI TS 103 666-2 [10], clause 7.2.

Req.ID	Clause	Description
	7.2.1	***Introduction***
RQ0702_001	7.2.1	The provisions of GlobalPlatform VPP - Concepts and Interfaces [15], clause 5.8 (without its subclauses) shall apply.
	7.2.2	***Kernel objects***
RQ0702_002	7.2.2	The provisions of GlobalPlatform VPP - Concepts and Interfaces [15], clause 5.8.1 shall apply.
	7.2.3	***Global requirements and mandatory Access Control rules***
RQ0702_003	7.2.3	The provisions of GlobalPlatform VPP - Concepts and Interfaces [15], clause 5.8.2 shall apply.
	7.2.4	***Process states diagram***
RQ0702_004	7.2.4	The provisions of GlobalPlatform VPP - Concepts and Interfaces [15], clause 5.8.3 shall apply.
	7.2.5	***Definition of the process states***
RQ0702_005	7.2.5	The provisions of GlobalPlatform VPP - Concepts and Interfaces [15], clause 5.8.4 shall apply.
	7.2.6	***Mandatory access control***
RQ0702_006	7.2.6	The provisions of GlobalPlatform VPP - Concepts and Interfaces [15], clause 5.11 shall apply.
RQ0702_007	7.2.6	The low-level operating system shall only have non-shareable memory regions.

5.5.3 Services

Reference: ETSI TS 103 666-2 [10], clause 7.3.

Req.ID	Clause	Description
	7.3.1	***Secondary Platform Bundle Loader***
RQ0703_001	7.3.1.1	The Primary Platform shall support a Secondary Platform Bundle Loader as defined in GlobalPlatform Open Firmware Loader for Tamper Resistant Element [13] with the exception that the OFL ARP state shall be UNLOCKED.
RQ0703_002	7.3.1.1	The Secondary Platform Bundle Loader shall be a system Secondary Platform Bundle and shall support the requirements defined in the augmented package loader 2 in BSI-CC-PP-0084-2014 [2].
RQ0703_003	7.3.1.2	The Secondary Platform Bundle Loader shall implement at least the registry entries of the OFL service gate as listed in Table 7.1: Registry entry in the OFL Service Gate of ETSI TS 103 666-2 [10].
RQ0703_004	7.3.1.2	The Secondary Platform Bundle Loader shall implement the registry entries of the OFL service gate as listed in Table 7.2: Additional registry entry in the OFL Service Gate of ETSI TS 103 666-2 [10].
RQ0703_005	7.3.1.2	If the iSSP contains or is intended to contain at least one Telecom Secondary Platform Bundle, TELECOM_CAPABILITY shall be set at the time of manufacturing. It shall contain the maximum number of distinct concurrent 3GPP network registrations based on different subscriber identifiers, supported by the terminal.
RQ0703_006	7.3.1.3	The Secondary Platform Bundle Loader shall support the commands defined in GlobalPlatform VPP - OFL VNP Extension [13].
RQ0703_007	7.3.1.3	The Secondary Platform Bundle Loader shall support the commands listed in Table 7.3: Additional commands supported by the OFL Service Gate of ETSI TS 103 666-2 [10].
RQ0703_008	7.3.1.4	The Secondary Platform Bundle Loader shall support the responses defined in GlobalPlatform VPP - OFL VNP Extension [13].
RQ0703_009	7.3.1.4	The OFL service gate shall support the responses listed in Table 7.4: Additional responses supported by the OFL Service Gate of ETSI TS 103 666-2 [10]. • eSPBL_E_NO_CI_FOR_SPBM_VERIFICATION

Req.ID	Clause	Description
RQ0703_010	7.3.1.4	The OFL service gate shall support the responses listed in Table 7.4: Additional responses supported by the OFL Service Gate of ETSI TS 103 666-2 [10]. • eSPBL_E_NO_CI_FOR_SPBL_VERIFICATION
RQ0703_011	7.3.1.4	The OFL service gate shall support the responses listed in Table 7.4: Additional responses supported by the OFL Service Gate of ETSI TS 103 666-2 [10]. • eSPBL_E_NO_CI_FOR_KEYAGREEMENT
RQ0703_012	7.3.1.4	The OFL service gate shall support the responses listed in Table 7.4: Additional responses supported by the OFL Service Gate of ETSI TS 103 666-2 [10]. • eSPBL_E_NO_SUPPORTED_CRYPTO
RQ0703_013	7.3.1.4	The OFL service gate shall support the responses listed in Table 7.4: Additional responses supported by the OFL Service Gate of ETSI TS 103 666-2 [10]. • eSPBL_E_INVALID_SPBM_CERT
RQ0703_014	7.3.1.4	The OFL service gate shall support the responses listed in Table 7.4: Additional responses supported by the OFL Service Gate of ETSI TS 103 666-2 [10]. • eSPBL_E_EXCEED_TELECOM_CAPABILITY
RQ0703_015	7.3.1.5	The Secondary Platform Bundle Loader shall manage firmware sessions as defined in GlobalPlatform Open Firmware Loader for Tamper Resistant Element [13], section 2.2.1 as per the Secondary Platform Bundles installed in the iSSP.
RQ0703_016	7.3.1.5	The Secondary Platform Bundle Loader shall manage the notification counter value as additional parameter of the firmware session. The 4 bytes integer value which is used when the Secondary Platform Bundle Loader generates the notification token as defined in clause 12.6.2.8 of ETSI TS 103 666-2 [10]. The notification counter of the Secondary Platform Bundle shall be pre-incremented by one by Secondary Platform Bundle Loader at each generation of a token. The initial value of the counter is '1'.
RQ0703_-017	7.3.1.5	The Secondary Platform Bundle Loader shall manage the Secondary Platform Bundle private identifier as defined in clause 9.4.5 of ETSI TS 103 666-2 [10].
RQ0703_018	7.3.1.5	The Secondary Platform Bundle Loader shall manage the SPB metadata as defined in clause 12.6.2.6 of ETSI TS 103 666-2 [10] for the Secondary Platform Bundle container. When the firmware session is created, the SPB metadata contained in the bound SPB image shall be stored.
RQ0703_019	7.3.1.5	The Secondary Platform Bundle Loader shall manage the SPB state as additional parameter of the firmware session providing the current state of the Secondary Platform Bundle. The value shall be one of 'Disabled (0)' and 'Enabled (1)'.
	7.3.2	***Communication service***
RQ0703_020	7.3.2	The Primary Platform shall provide communication service for the use of the Secondary Platform Bundle to communicate with entities outside the iSSP. The interface is defined in clause 8.2.
	7.3.3	***Management service***
RQ0703_021	7.3.3	The Primary Platform shall provide management service for the exclusive use of the Secondary Platform Bundle Loader.
RQ0703_022	7.3.3	The management service provides the interface to manage: • the life cycle of a Secondary Platform Bundle.
RQ0703_023	7.3.3	The management service provides the interface to manage: • the installation and management of a Secondary Platform Bundle by a Secondary Platform Bundle Loader.

5.5.4 Minimum level of interoperability

Reference: ETSI TS 103 666-2 [10], clause 7.4.

Req.ID	Clause	Description
RQ0704_001	7.4	The provisions of GlobalPlatform VPP - Concepts and Interfaces [15], clause 7 and its subclauses shall apply.

5.5.5 Primary Platform identification

Reference: ETSI TS 103 666-2 [10], clause 7.5.

Req.ID	Clause	Description
RQ0705_001	7.5	The Primary Platform instance is identified by a Primary Platform identifier. The Primary Platform identifier is a sequence of 32 characters, divided in 8 groups of 4 characters each, with a dash between each group.
RQ0705_002	7.5	The Primary Platform identifier shall not be changed irrespective of the Firmware update of the Secondary Platform Bundle Loader.

5.5.6 Provisioning of Primary Platform software

Reference: ETSI TS 103 666-2 [10], clause 7.6.

Req.ID	Clause	Description
RQ0706_001	7.6	The provisions of GlobalPlatform VPP - Concepts and Interfaces [15], clause 5.7 REQ85 shall apply.

5.5.7 Part Number Identifier

Reference: ETSI TS 103 666-2 [10], clause 7.7.

Req.ID	Clause	Description
RQ0707_001	7.7	It shall be possible to retrieve the Primary Platform manufacturer, the model of the Primary Platform, the assurance level of the Primary Platform and the Secondary Platform Bundle Loader using the Part Number identifier.

5.6 Primary Platform Interface

Reference: ETSI TS 103 666-2 [10], clause 8.

Req.ID	Clause	Description
	8.1	***Kernel functions ABI/API***
RQ0801_001	8.1	The provisions of GlobalPlatform VPP - Concepts and Interfaces [15], clause 5.8.5 shall apply.
	8.2	***Communication service interface***
RQ0802_001	8.2	The provisions of GlobalPlatform VPP - Concepts and Interfaces [15], clause 5.9 shall apply.
	8.3	***Secondary Platform Bundle management service interface***
RQ0803_001	8.3	The provisions of GlobalPlatform VPP - Concepts and Interfaces [15], clause 5.10 and its subclauses shall apply with the exception that the states Active and Deleted as defined in clause 9.2 of ETSI TS 103 666-2 [10] are not relevant for this Secondary Platform Bundle management service interface.

5.7 Secondary Platform Bundle

Reference: ETSI TS 103 666-2 [10], clause 9.

Req.ID	Clause	Description
	9.2	***States***
RQ0902_001	9.2	The states and transitions of the Secondary Platform Bundle container shall be as defined in GlobalPlatform VPP - Concepts and Interfaces [15], clause 5.10 and its subclauses.
RQ0902_002	9.2	The state of the Secondary Platform Bundle instance shall be as defined in GlobalPlatform VPP - Concepts and Interfaces [15], clause 6.3.
RQ0902_003	9.2	The Secondary Platform Bundle Loader manages the states of the Secondary Platform Bundle container, and not the states of the Secondary Platform Bundle instance.

Req.ID	Clause	Description
	9.2	***States***
RQ0902_004	9.2	The mechanism to make sure that the Secondary Platform Bundle Loader can disable an active Secondary Platform Bundle is the host arbitration process described in ETSI TS 103 666-1 [9], clause 8.2.
RQ0902_005	9.2	The context of the Secondary Platform Bundle and of its applications is valid only in the Enabled and Active states. The context is created when the Secondary Platform Bundle moves to Active state for the first time.
RQ0902_006	9.2	If the state of a Secondary Platform Bundle changes from Disabled to Enabled, upon the next state change to Active, the Secondary Platform Bundle shall erase its context and the context of its SSP Applications.
RQ0902_007	9.2	No more than a single Secondary Platform Bundle shall be in an Active state.
	9.3	***Secondary Platform Bundle container format***
RQ0903_001	9.3	The provisions of GlobalPlatform VPP - Firmware Format [14] shall apply.
	9.4	***Secondary Platform***
	9.4.1	***High Level OS***
RQ0904_001	9.4.1	The Secondary Platform Bundle contains a High Level OS as defined in GlobalPlatform VPP - Concepts and Interfaces [15], clause 6.4 and its subclauses.
	9.4.2	***Execution Framework***
RQ0904_002	9.4.2	The provisions of ETSI TS 103 666-1 [9], clause 5.4 shall apply.
	9.4.3	***UICC platform as a Secondary Platform***
RQ0904_003	9.4.3	The Secondary Platform may emulate a UICC platform as defined in ETSI TS 102 221 [4] and ETSI TS 102 223 [5]. In this case the Secondary Platform shall support the UICC APDU gate, as described in ETSI TS 103 666-1 [9], clause 10.2.8.2 and the UICC specific mechanisms defined in ETSI TS 103 666-1 [9], clauses 5.5, 6.6.1, 6.8.1, 6.10 and 10.2.
	9.4.4	***Capability exchange***
RQ0904_004	9.4.4	The data field sent by the terminal to the iSSP during the capability exchange procedure shall contain the data structure defined in ETSI TS 103 666-1 [9], clause 6.4.2.4, with the following modifications: • aPhysicalInterfaces in the TerminalCapability should not be present.
RQ0904_005	9.4.4	The data field sent by the terminal to the iSSP during the capability exchange procedure shall contain the data structure defined in ETSI TS 103 666-1 [9], clause 6.4.2.4, with the following modifications: • aPhysicalInterfaces in the TerminalCapability shall be ignored if present.
RQ0904_006	9.4.4	The data field sent by the iSSP to the terminal during the capability exchange procedure shall contains the data structure defined in ETSI TS 103 666-1 [9], clause 6.4.2.5, with the following modifications: • SSPClass field shall have the eSSPClass-Integrated (0) value.
RQ0904_007	9.4.4	The data field sent by the iSSP to the terminal during the capability exchange procedure shall contains the data structure defined in ETSI TS 103 666-1 [9], clause 6.4.2.5, with the following modifications: • aClassSpecificCapabilities, aPhysicalInterfaces; and • aSspExternalMaxPowerConsumption in the SSPCapability should not be present.
RQ0904_008	9.4.4	The data field sent by the iSSP to the terminal during the capability exchange procedure shall contains the data structure defined in ETSI TS 103 666-1 [9], clause 6.4.2.5, with the following modifications: • aClassSpecificCapabilities, aPhysicalInterfaces; and • aSspExternalMaxPowerConsumption in the SSPCapability shall be ignored if present.
	9.4.5	***Identifiers of Secondary Platform Bundle***
RQ0904_009	9.4.5	The Secondary Platform Bundle shall be provided with two identifiers: 1) The Secondary Platform Bundle identifier (i.e. SpbId), UUID which is the same as the public UUID of a firmware as defined in GlobalPlatform Open Firmware Loader for Tamper Resistant Element [13].
RQ0904_010	9.4.5	The Secondary Platform Bundle shall be provided with two identifiers: 2) The Secondary Platform Bundle private identifier (i.e. PrivateSpbId), UUID which is the same as the private UUID of a firmware as defined in GlobalPlatform Open Firmware Loader for Tamper Resistant Element [13]; where the NSS used to create the UUID shall have an entropy in its of at least 32 bytes.

Req.ID	Clause	Description
	9.5	***SSP Application***
	9.5.2	***Lifecycle management***
RQ0905_001	9.5.2	After the Secondary Platform Bundle has been loaded by the Secondary Platform Bundle Loader, the Secondary Platform Bundle and its SSP Applications are in their initial lifecycle state: • for a Secondary Platform Bundle supporting the legacy execution framework defined in ETSI TS 102 241 [7], the initial lifecycle state is determined by the Secondary Platform Bundle maker; • for a Secondary Platform Bundle supporting a native application, the initial lifecycle state is determined by the Secondary Platform Bundle maker; • for a Secondary Platform Bundle supporting a new type of execution framework, the initial lifecycle state is for future study.
RQ0905_002	9.5.2	After the Secondary Platform Bundle has been enabled by the Secondary Platform Bundle Loader, the Secondary Platform Bundle internal lifecycle states and their management shall be governed by the following rules: • for a Secondary Platform Bundle supporting the legacy framework defined in ETSI TS 102 241 [7], the rules and mechanisms for the management of the lifecycle of the Security Domains and Applications shall be compliant with the GlobalPlatform Card Specification [12] and ETSI TS 102 226 [6]; • for a Secondary Platform Bundle supporting native SSP Applications, the rules and mechanisms for the management of the lifecycle state of the SSP application(s) are proprietary and out of the scope of the present document; • for a Secondary Platform Bundle supporting a new type of execution framework, the rules and mechanisms for the management of the lifecycle states of the SSP application(s) are for future study.
	9.6	***Lifecycle management of Secondary Platform Bundles***
RQ0906_001	9.6	The Secondary Platform Bundle Loader shall enforce that the number of Telecom Secondary Platform Bundles in the Enabled or Active state is not greater than the TELECOM_CAPABILITY parameter value defined in clause 7.3.1 of ETSI TS 103 666-2 [10].
	9.7	***Secondary Platform Bundle family identifier***
RQ0907_001	9.7	A family of Secondary Platform Bundles is identified by a family identifier. A family identifier is a UUID computed from a URN using IETF RFC 4122 UUID version 5 [19]. It is equivalent to the Firmware Family in GlobalPlatform Open Firmware Loader for Tamper Resistant Element [13].

5.8 Communication interface

Reference: ETSI TS 103 666-2 [10], clause 10.

Req.ID	Clause	Description
	10.2	***SSP Common Layer***
RQ1002_001	10.2	The iSSP shall support the SCL protocol layer, as defined in ETSI TS 103 666-1 [9], clause 8.
RQ1002_002	10.2	The SCL router and the network controller host shall share the same security perimeter as the Primary Platform in order to ensure the correct mapping of host aliases with the corresponding SCL host.
RQ1002_003	10.2	Each Secondary Platform Bundle is responsible for the implementation of the SCL protocol as needed for its operation.

5.9 Certification

Reference: ETSI TS 103 666-2 [10], clause 11.

Req.ID	Clause	Description
	11.1	***Introduction***
RQ1101_001	11.1	The iSSP shall be able to support a certification by composition of a Secondary Platform Bundle from the Primary Platform certification.
	11.2	***Primary Platform certification***
	11.2.1	***Overview***
RQ1102_001	11.2.1	The provisions of GlobalPlatform VPP - Concepts and Interfaces [15], clause 4 shall apply.
RQ1102_002	11.2.1	The certification of the Primary Platform shall include the Loader Package 2, as defined in BSI-CC-PP-0084-2014 [2].

5.10 iSSP ecosystem and interfaces

5.10.1 Security overview

Reference: ETSI TS 103 666-2 [10], clause 12.2.

Req.ID	Clause	Description
	12.2.1	***Public key infrastructure for Si4 interface***
RQ1202_001	12.2.1.1.1.1	The Secondary Platform Bundle Loader certification path for digital signature shall include the following certificates: • CI certificate. • SSP maker certificate. • SPBL certificate: The SPBL certificate shall contain the public key used to verify the signature generated by the Secondary Platform Bundle Loader.
RQ1202_002	12.2.1.1.1.1	The Secondary Platform Bundle Loader certification path may include the following certificates: • CI subordinate CA certificate: The CI subordinate CA certificate shall be issued by a CI. The CI subordinate CA certificate can be used to verify an SSP maker certificate. • SSP maker subordinate CA certificate: The SSP maker subordinate CA certificate shall be issued by an SSP maker. The SSP maker subordinate CA certificate can be used to verify an SPBL certificate.
RQ1202_003	12.2.1.1.1.2	The SPBM certification path for digital signature shall include the following certificates: • CI certificate. • SPBM DS certificate: The SPBM DS certificate shall be used to verify the signature generated by the SPB Manager.
RQ1202_004	12.2.1.1.1.2	The SPBM certification path for key agreement shall include the following certificates: • CI certificate. • SPBM KA certificate: The SPBM KA certificate shall be used to generate a session key for secure communication between the SPB Manager and the Secondary Platform Bundle Loader.
RQ1202_005	12.2.1.1.1.2	The SPBM certification path for digital signature and key agreement may include the following certificates: • CI subordinate CA certificate: The CI subordinate CA certificate shall be issued by a CI. The CI subordinate CA certificate can be used to verify an SPBM Subordinate CA certificate. The CI subordinate CA certificate can be used to verify an SPBM DS certificate and SPBM KA certificate. • SPBM subordinate CA certificate: The SPBM subordinate CA certificate shall be issued by an SPBM or CI. The SPBM subordinate CA certificate can be used to verify an SPBM DS certificate and SPBM KA certificate.
RQ1202_006	12.2.1.1.2.1	Basic certificate fields for all certificates used by the Secondary Platform Bundle Loader and the SPB Manager are identified in Table 12.2 of ETSI TS 103 666-2 [10] and follow the X.509 v3 certificate format as defined in IETF RFC 5280 [21].
RQ1202_007	12.2.1.1.2.2	The Authority key identifier (IETF RFC 5280 [21], section 4.2.1.1) is a considered extension field for Certificates: • All the certificate except for CI certificate shall contain the extension for authority key identifier.

Req.ID	Clause	Description
RQ1202_008	12.2.1.1.2.2	The Subject key identifier (IETF RFC 5280 [21], section 4.2.1.2) is a considered extension field for Certificates: • All the certificate shall contain the extension for subject key identifier. The value of this field shall be the identifier of the public key contained in the certificate.
RQ1202_009	12.2.1.1.2.2	The Key usage (IETF RFC 5280 [21], section 4.2.1.3) is a considered extension field for Certificates: • For a certificate used for verifying its subject certificate, keyCertSign (bit 5) shall be asserted to the key usage extension field of the certificate. • For the last certificate in the Secondary Platform Bundle Loader certification path for signature generation, digitalSignature (bit 0) shall be asserted to the key usage extension. • For the last certificate in the SPB Manager certification path for key agreement, key Agreement (bit 4) bit shall be asserted to the key usage extension.
RQ1202_010	12.2.1.1.2.2	Certificate polices (IETF RFC 5280 [21], section 4.2.1.4) are a considered extension field for Certificates: • Each certificate shall have the appropriate value of the extension for certificate policies. • The OIDs used for value of the extension for certificate polices are defined as follows: – SubjectAltName (IETF RFC 5280 [21], section 4.2.1.6): A certificate may have the extension forsubjectAltName. – Basic constraints (IETF RFC 5280 [21], section 4.2.1.9): For any CA or subordinate CA certificate, the value of the extension for basic constraints shall be asserted.
RQ1202_011	12.2.1.1.2.2	The following additional extension fields shall be present according to the following rules: • Primary Platform identifier: The Primary Platform identifier extension field shall only be contained in the SPBL certificate.
RQ1202_012	12.2.1.1.2.2	The following additional extension fields shall be present according to the following rules: • The Primary Platform identifier extension field shall contain the Primary Platform identifier of the SSP.
RQ1202_013	12.2.1.1.2.2	The following additional extension fields shall be present according to the following rules: • Family identifier: A certificate shall contain the family identifier extension field if it is present in its parent-certificate. If it is not present in the parent-certificate, a certificate may contain the family identifier extension field. The family identifier extension field shall indicate the list of family identifiers associated with the certification path to load a Secondary Platform Bundle image. If the family identifier extension field is present in the parent certificate, this list may contain the list or a subset of the list of family identifiers indicated in the family identifier extension field of the parent certificate.
RQ1202_014	12.2.1.1.2.2	The following additional extension fields shall be present according to the following rules: • Custodian identifier: A certificate shall contain the custodian identifier extension field if it is present in its parent-certificate. If it is not present in the parent-certificate, a certificate may contain the custodian identifier extension field. The custodian identifier extension field shall indicate the list of OIDs of custodians associated with the certification path to load a Secondary Platform Bundle image. If the custodian identifier extension field is present in the parent certificate, this list may contain the list or a subset of the list of custodian identifiers indicated in the custodian identifier extension field of the parent certificate as defined in table 12.3 of ETSI TS 103 666-2 [10].
RQ1202_015	12.2.1.1.3	For 'subjectPublicKeyInfo' field, the following settings shall apply: • 'AlgorithmIdentifier.algorithm' field shall be set to: – if the value of 'Extension for KeyUsage' field is set to digitalSignature(0) and/or keyCertSign(5); – for Elliptic Curve Digital Signature Algorithm (ECDSA), "iso(1) member-body(2) us(840) ansi-X9-62(10045) keyType(2) ecPublicKey(1)" as defined in IETF RFC 5480 [22].
RQ1202_016	12.2.1.1.3	For 'subjectPublicKeyInfo' field, the following settings shall apply: • 'AlgorithmIdentifier.algorithm' field shall be set to: – if the value of 'Extension for KeyUsage' field is set to digitalSignature(0) and/or keyCertSign(5); – for SM2 digital signature algorithm, "iso(1) standard(0) digital-signature-with-appendix(14888) part3(3) algorithm(0) sm2(14)" as defined in ISO/IEC 14888-3 [29].

Req.ID	Clause	Description
RQ1202_017	12.2.1.1.3	For 'subjectPublicKeyInfo' field, the following settings shall apply: • 'AlgorithmIdentifier.algorithm' field shall be set to: – if the value of 'Extension for KeyUsage' field is set to keyAgreement(4): – for Elliptic Curve Diffie-Hellman (ECDH), "iso(1) identified-organization(3) certicom(132) schemes (1) ecdh(12)" as defined in IETF RFC 5480 [22].
RQ1202_018	12.2.1.1.3	For 'subjectPublicKeyInfo' field, the following settings shall apply: • 'AlgorithmIdentifier.parameters' field shall be set to: – for BrainpoolP256r1: "iso(1) identified-organization(3) teletrust(36) algorithm(3) signatureAlgorithm(3) ecSign(2) ecStdCurvesAndGeneration(8) ellipticCurve(1) versionOne(1) brainpoolP256r1(7)" as defined in IETF RFC 5639 [23].
RQ1202_019	12.2.1.1.3	For 'subjectPublicKeyInfo' field, the following settings shall apply: • 'AlgorithmIdentifier.parameters' field shall be set to: – for BrainpoolP384r1: "iso(1) identified-organization(3) teletrust(36) algorithm(3) signatureAlgorithm(3) ecSign(2) ecStdCurvesAndGeneration(8) ellipticCurve(1) versionOne(1) brainpoolP384r1(11)" as defined in IETF RFC 5639 [23].
RQ1202_020	12.2.1.1.3	For 'subjectPublicKeyInfo' field, the following settings shall apply: • AlgorithmIdentifier.parameters' field shall be set to: – for NIST P-256: "iso(1) member-body(2) us(840) ansi-X-9-62(10045) curves(3) prime(1) secp256v1(7)" as defined in IETF RFC 5480 [22].
RQ1202_021	12.2.1.1.3	For 'subjectPublicKeyInfo' field, the following settings shall apply: • 'AlgorithmIdentifier.parameters' field shall be set to: – for NIST P-384: "iso(1) identified-organization(3) certicom(132) curve(0) secp384r1(34)" as defined in IETF RFC 5480 [22].
RQ1202_022	12.2.1.1.3	For 'signature' and 'signatureAlgorithm' fields, the following settings shall apply: • 'AlgorithmIdentifier.algorithm' field shall be set to: – for ECDSA-with-SHA256: "iso(1) member-body(2) us(840) ansi-X9-62(10045) signatures(4) ecdsawith-SHA2(3) ecdsa-with-SHA256(2)" as defined in IETF RFC 5758 [25].
RQ1202_023	12.2.1.1.3	For 'signature' and 'signatureAlgorithm' fields, the following settings shall apply: • 'AlgorithmIdentifier.algorithm' field shall be set to: – for ECDSA-with-SHA384: "iso(1) member-body(2) us(840) ansi-X9-62(10045) signatures(4) ecdsawith-SHA2(3) ecdsa-with-SHA384(3)" as defined in IETF RFC 5758 [25].
RQ1202_024	12.2.1.1.3	For 'signature' and 'signatureAlgorithm' fields, the following settings shall apply: • 'AlgorithmIdentifier.algorithm' field shall be set to: – for SM2 digital signature algorithm, "iso(1) standard(0) digital-signature-with-appendix(14888) part3(3) algorithm(0) sm2(14)" as defined in ISO/IEC 14888-3 [29].
RQ1202_025	12.2.1.1.3	For 'signature' and 'signatureAlgorithm' fields, the following settings shall apply: • 'AlgorithmIdentifier.parameters' field shall be set to: – for ECDSA-with-SHA256 and ECDSA-with-SHA384: the parameters field shall be omitted as defined in IETF RFC 5754 [24], section 3.2.
RQ1202_026	12.2.1.1.4	Both the Secondary Platform Bundle Loader and the SPB Manager shall verify the certificate chain received by each other.
RQ1202_027	12.2.1.1.4	A certificate chain to be verified shall satisfy the following conditions: • The value of Authority Key Identifier extension field in a subject's certificate shall be the same as the value of Subject Key Identifier extension field in the issuer's certificate which is used to verify the subject's certificate. • The value of issuer field in a subject's certificate shall be the same as the value of subject field in the issuer's certificate which is used to verify the subject's certificate. • All certificates in the certificate chain shall use the same digital signature algorithm and parameter set indicated by one of the algorithm Identifiers defined in clause 12.2.1.1.3 of ETSI TS 103 666-2 [10]. • All certificates in the certificate chain shall not have been expired. • All certificates in the certificate chain shall not have been revoked. The certificate revocation status shall be checked as defined in clause 12.2.1.1.5 of ETSI TS 103 666-2 [10].
RQ1202_028	12.2.1.1.4	The Secondary Platform Bundle Loader and the SPB Manager shall manage the trusted Public Key information to verify received certificate chain.

Req.ID	Clause	Description
RQ1202_029	12.2.1.1.4	Each set of trusted public key information shall contain the following: • Public Key information (SubjectPublicKeyInfo as defined in IETF RFC 5280 [21]). • Subject Key Identifier of the Public Key. • Family identifier(s) associated with that Public Key, if any. • Custodian identifier(s) associated with that Public Key, if any.
RQ1202_030	12.2.1.1.4	The Secondary Platform Bundle Loader and the SPB Manager shall verify the received certificate chain by using one set of trusted Public Key information as the trust anchor information for the certification path validation procedure defined in IETF RFC 5280 [21]. The trust anchor shall be determined during the download procedure as defined in clause 12.3.3 of ETSI TS 103 666-2 [10].
RQ1202_031	12.2.1.1.4	The Secondary Platform Bundle Loader and the SPB Manager shall: • Perform the certification path validation defined in IETF RFC 5280 [21] to verify the received certificate chain. • Verify that the received certificate chain follows one of the certificate chains as defined in clause 12.2.1.1.1 of ETSI TS 103 666-2 [10]. • Verify that all certificate(s) in the received certificate chain follow the corresponding certificate description(s) as defined in clause 12.2.1.1.2 of ETSI TS 103 666-2 [10].
RQ1202_032	12.2.1.1.4	In addition, the Secondary Platform Bundle Loader and the SPB Manager shall verify that the certificates in the received certificate chain satisfy the condition as described below: • If a certificate contains the list of family identifier(s) in the family identifier extension field, the list shall contain the family identifier associated with the trust anchor used for the certification path validation. • If a certificate contains the list of custodian identifier(s) in the custodian identifier extension field, the list shall contain the custodian identifier associated with the trust anchor used for the certification path validation.
RQ1202_033	12.2.1.1.4	If any of the verifications described above fails, the certificate chain shall be considered as invalid.
RQ1202_034	12.2.1.1.5	The Secondary Platform Bundle Loader and the SPB Manager may verify the revocation status of certificates in the received certification path by using a Certificate Revocation List (CRL) as defined in IETF RFC 5280 [21].
RQ1202_035	12.2.1.1.5	The SPB Manager may verify the revocation status of certificates in the received certification path by using a Certificate Revocation List (CRL) as defined in IETF RFC 5280 [21].
RQ1202_036	12.2.1.1.5	The SPB Manager may obtain a CRL by accessing a public repository with the uniform resource identifier indicated in cRLDistributionPoint extension of a certificate as defined in IETF RFC 5280 [21].
RQ1202_037	12.2.1.1.5	The Secondary Platform Bundle Loader may obtain a CRL with the following mechanism: • The SPB Manager may provide the list of the latest CRL for each certificate in its certification path along with the certification path. The Secondary Platform Bundle Loader shall verify the revocation status of the certificates with the corresponding CRLs provided by the SPB Manager.
	12.2.2	***Cryptographic algorithms***
RQ1202_038	12.2.2.1	The Secondary Platform Bundle Loader and the SPB Manager shall support at least one out of the following elliptic curve domain parameter sets: • NIST P-256 as defined in NIST 800-56A [31]. • NIST P-384 as defined in NIST 800-56A [31]. • brainpoolP256r1 as defined in IETF RFC 5639 [23]. • brainpoolP384r1 as defined in IETF RFC 5639 [23].
RQ1202_039	12.2.2.2	The Secondary Platform Bundle Loader and the SPB Manager shall support at least one of the following algorithms to generate and verify signatures: • Elliptic Curve Digital Signature Algorithm (ECDSA) as defined in ANSI X9.62-2005 [1].
RQ1202_040	12.2.2.2	The Secondary Platform Bundle Loader and the SPB Manager shall support at least one of the following algorithms to generate and verify signatures: • SM2 digital signature algorithm as defined in ISO/IEC 14888-3 [29]. The hash function shall be the SM3 hash function as defined in ISO/IEC 10118-3 [28].

Req.ID	Clause	Description
RQ1202_041	12.2.2.3	The Secondary Platform Bundle Loader and the SPB Manager shall support at least one of the following key agreement algorithms to establish session keys: • Elliptic Curve Key Agreement Algorithm (ECKA) as defined in BSI TR-03111 [3]. • SM2 Key exchange Algorithm (SM2KA) as defined in SM2 Digital Signature Algorithm [17], section 6.2. If ECKA is used as the key agreement algorithm, a shared secret shall be generated by using one between either the ElGamal key agreement protocol or the Diffie-Hellman key agreement protocol. The session key shall be computed from the generated shared secret value using the X9.63 key derivation function as described in GP Open Firmware Loader for Tamper Resistant Element [13], clause 3.2.1.
RQ1202_042	12.2.2.4	The Secondary Platform Bundle Loader and the SPB Manager shall support at least one of the following block cipher algorithms to encrypt data: • eGCM based on AES-128 as defined in GP Open Firmware Loader for Tamper Resistant Element [13], Annex A. • eGCM based on AES-256 as defined in GP Open Firmware Loader for Tamper Resistant Element [13], Annex A.

5.10.2 Secondary Platform Bundle provisioning procedure

Reference: ETSI TS 103 666-2 [10], clause 12.3.

Req.ID	Clause	Description
	12.3.1	***Overview***
RQ1203_001	12.3.1	The preparation procedure shall be performed between a Service Provider and an SPB Manager over the Si1 interface and a Service Provider and a Subscriber. The Subscriber is able to obtain from the service provider the relevant information for loading a Secondary Platform Bundle, including the URL of the SPB Manager to which the iSSP shall establish a secure communication channel to start the download procedure.
RQ1203_002	12.3.1	The download procedure shall be performed between a Secondary Platform Bundle Loader and an SPB Manager over the Si2 and Si3 interfaces. The bound Secondary Platform Bundle image shall be delivered securely from the SPB Manager to the LBA through the download procedure.
RQ1203_003	12.3.1	The installation procedure shall be performed between an LBA and a Secondary Platform Bundle Loader over the Si3 interface. The bound Secondary Platform Bundle image shall be delivered from the LBA to the Secondary Platform Bundle Loader through the installation procedure. The Secondary Platform Bundle Loader installs the Secondary Platform Bundle container by decrypting the bound Secondary Platform Bundle image.
RQ1203_004	12.3.1	The notification procedure shall be performed between the Secondary Platform Bundle Loader and the SPB Manager over the Si1, Si2 and Si3 interfaces with either: • Case 1: the preparation procedure is completed before the download procedure is triggered.
RQ1203_005	12.3.1	The notification procedure shall be performed between the Secondary Platform Bundle Loader and the SPB Manager over the Si1, Si2 and Si3 interfaces or: • Case 2: the preparation procedure is not completed before the download procedure is triggered, i.e. the service provider needs additional information from the terminal and/or the SSP to select the Secondary Platform Manager.
	12.3.2	***Preparation procedure***
RQ1203_006	12.3.2.1	The preparation procedure consists of the following two processes: • Secondary Platform Bundle selection process: the selection of the Secondary Platform Bundle allowing the service provider to select a Secondary Platform Bundle that matches the terminal and the SSP capabilities. • Service provider reference creation process: the creation of a reference shared between the service provider and the SPB Manager. This allows the end user to trigger the download procedure as defined in clause 12.3.3 of ETSI TS 103 666-2 [10]. The Secondary Platform Bundle selection process and the service provider reference creation process may be executed in any order or in once, according to the service provider's implementation choices.
RQ1203_007	12.3.2.2	The selection of the Secondary Platform Bundle allows the service provider to select a Secondary Platform Bundle that matches the terminal and the SSP capabilities. This is performed using the "Si1.SelectSpb" function. This process may be executed regardless of the service provider reference creation process defined in clause 12.3.2.3 of ETSI TS 103 666-2 [10].

Req.ID	Clause	Description
RQ1203_008	12.3.2.2	If a CodeM is provided as input parameter of the "Si1.SelectSpb" function and is known by the SPB Manager and not already linked to another Secondary Platform Bundle, the service provider reference creation process is not needed.
RQ1203_009	12.3.2.2	If the service provider has set aFlagFinalize to TRUE in the "Si1.SelectSpb" function command, the SPB Manager shall wait for the completion of the Secondary Platform Bundle selection process (i.e. after it has sent the response to the "Si1.FinalizePreparation" function related to this Secondary Platform Bundle) to continue with the Bound SPB image download as defined in clause 12.3.3.2 of ETSI TS 103 666-2 [10].
RQ1203_010	12.3.2.3	The service provider reference creation process shall be performed using the "Si1.CreateSPReference" function and may be executed regardless of the Secondary Platform Bundle selection process defined in clause 12.3.2.2 of ETSI TS 103 666-2 [10].
RQ1203_011	12.3.2.3	The service provider may pass the CodeM value to be used as a parameter of the "Si1.CreateSPReference" function command. If this is not present, the SPB Manager shall generate the CodeM.
RQ1203_011a	12.3.2.3	The service provider may pass the CodeM value to be used as a parameter of the "Si1.CreateSPReference" function command.
RQ1203_011b	12.3.2.3	If service provider is not passing over the CodeM value this is not present, the SPB Manager shall generate the CodeM.
RQ1203_012	12.3.2.3	The service provider may pass the CodeM value to be used as a parameter of the "Si1.CancelPreparation" function command. This procedure allows the service provider to cancel a pending preparation procedure.
	12.3.3	***Download procedure***
RQ1203_013	12.3.3.1	The following shall be determined by the capability negotiation procedure: • SPBM certificate for key agreement and its certification path. • Public Key provisioned on the SPB Manager to verify the SPBL certificate and its certification path. • Data encryption algorithm used by the SPB Manager and the Secondary Platform Bundle Loader.
RQ1203_014	12.3.3.1	The LBA shall obtain the address of the SPB Manager (spbmAddr) e.g. from the end user.
RQ1203_015	12.3.3.1	The LBA may obtain the family identifier of the Secondary Platform Bundle container (spbFamilyId) to load. If the family identifier is present, the LBA may also obtain the OID of a custodian of that family identifier. The Secondary Platform Bundle Loader shall have the following: • Private key(s) for creating the Secondary Platform Bundle Loader signature. • Secondary Platform Bundle Loader certificate(s) for the digital signature used to verify the Secondary Platform Bundle Loader signature. • Secondary Platform Bundle Loader certificate chain(s) to be used by an SPB Manager for verifying Secondary Platform Bundle Loader certificate for digital signature. • Trusted public key(s) and algorithmIdentifier value(s) to be used to verify certificate(s) from an SPB Manager as per the family identifier(s) and/or custodian(s). • List of supported algorithmIdentifier value(s) for key agreement and data encryption.
RQ1203_016	12.3.3.1	The SPB Manager shall have the following: • Private key(s) for creating the SPB Manager signature. • Private key(s) for key agreement. • SPB Manager certificate(s) for digital signature used to verify the SPB Manager signature. • SPB Manager certificate(s) for key agreement. • SPB Manager certificate chain(s) to be used by a Secondary Platform Bundle Loader for verifying the SPB Manager certificates for key agreement and for digital signature. • Trusted public key(s) and algorithmIdentifier value(s) to be used to verify the certificate(s) from the Secondary Platform Bundle Loader as per the family identifier(s) and/or custodian(s).
RQ1203_017	12.3.3.1	The capability negotiation procedure shall use the following steps: 1) The LBA shall call the "Si3.GetSspInfo" function. The function command may contain the spbFamilyId. If spbFamilyId is present, the function command may also contain the OID of the custodian for this spbFamilyId.

Req.ID	Clause	Description
RQ1203_018	12.3.3.1	The capability negotiation procedure shall use the following steps: 2) On reception of the "Si3.GetSspInfo" function command, the Secondary Platform Bundle Loader shall build aSspInfoPublic as defined in clause 12.6.2.2.2 of ETSI TS 103 666-2 [10]: a) aSpblSpecVerInfo; b) aSspGeneralCryptoInfo; and/or c) aSspFamilyCryptoInfoBlock, that may contain multiple SspFamilyCryptoInfoBlock data structures.
RQ1203_019	12.3.3.1	Each aSspFamilyCryptoInfoBlock data structure shall contain a family identifier and may contain the aSspFamilyCryptoInfo and/or the set of SspOidCryptoInfoBlock data structures.
RQ1203_020	12.3.3.1	If aSspFamilyCryptoInfoBlock data structure contains a SspOidCryptoInfoBlock data structure it shall contain the aCustodianOid and aSspOidCryptoInfo.
RQ1203_021	12.3.3.1	The aSspGeneralCryptoInfo, the aSspFamilyCryptoInfo and the aSspOidCryptoInfo shall comprise the following: • aSspPkIdForSpbmVerification: trusted public key identifier(s) available for the Secondary Platform Bundle Loader to verify SPB Manager certificate chain. • aSspPkIdForSpblVerification: trusted public key identifier(s). The SPB Manager shall use one of these trusted public key identifiers to verify Secondary Platform Bundle Loader certificate chain. • aKeyAgreementAlgorithmList: the list of algorithm identifiers for key agreement algorithms supported by the Secondary Platform Bundle Loader. • aCipherAlgorithmList: the list of algorithm identifiers of data encryption algorithms supported by the Secondary Platform Bundle Loader.
RQ1203_022	12.3.3.1	The capability negotiation procedure shall use the following steps: 3) The Secondary Platform Bundle Loader shall return the aSspInfoPublic to the LBA.
RQ1203_023	12.3.3.1	The capability negotiation procedure shall use the following steps: 4) The LBA shall establish a TLS connection with the SPB Manager in server authentication mode. NOTE: The establishment and management of the TLS connection are outside the scope of the present document.
RQ1203_024	12.3.3.1	The capability negotiation procedure shall use the following steps: 5) The LBA shall call the "Si2.GetSpbmCertificate" function with its input data comprising aSspInfoPublic and aTerminalInfo.
RQ1203_025	12.3.3.1	The capability negotiation procedure shall use the following steps: 6) On reception of "Si2.GetSpbmCertificate" function command, the SPB Manager shall verify that it supports the contents in aSspInfoPublic and aTerminalInfo.
RQ1203_026	12.3.3.1	The capability negotiation procedure shall use the following steps: 7) Using aSspInfoPublic, the SPB Manager shall choose: a) An SPB Manager certificate for key agreement that can be verified by the trusted public key indicated by one of the trusted public key identifiers in the aSspPkIdListForSpbmVerification. The algorithmIdentifier of the selected certificate shall be one of the algorithmIdentifier in aKeyAgreementAlgorithmList. b) An SPB Manager certificate for digital signature that can be verified by the trusted public key indicated by one of the trusted public key identifiers in the aSspPkIdListForSpbmVerification. c) One of trusted public key identifier(s) in the aSspPkIdListForSpblVerification that shall be used by the Secondary Platform Bundle Loader to select its certificate(s). The SPB Manager shall set the selected trusted public key identifier into aSspCiPkIdToBeUsed. d) One of algorithmIdentifiers in the aCipherAlgorithmList that shall be used by the Secondary Platform Bundle Loader and the SPB Manager for data encryption. The SPB Manager shall set the selected algorithmIdentifier into aSspCryptoToBeUsed. The SPB Manager shall generate a random challenge (ChallengeS) which is used in the authentication of the Secondary Platform Bundle Loader. If the SPB Manager cannot find the appropriate certificate(s), trusted public key identifier, or algorithmIdentifier value, the SPB Manager shall return an error to the LBA and shall terminate the procedure. NOTE: The handling of aTerminalInfo is outside the scope of the present document and may be defined by other organizations.
RQ1203_027	12.3.3.1	The capability negotiation procedure shall use the following steps: 8) The SPB Manager shall return the SPB Manager certificate for key agreement, aSspCiPkIdToBeUsed, aSspCryptoToBeUsed and aSpbFamilyId, and optionally the certificate chain for SPB Manager certificate for key agreement and the OID of a custodian of the aSpbFamilyId to the LBA.

Req.ID	Clause	Description
RQ1203_028	12.3.3.2	The bound SPB image download procedure shall use the following steps: 1) The LBA shall call the "Si3.SetSpbmCredential" function. The function command shall contain the index of IDS_CREDENTIAL_PARAMETER registry and the aSpbmCredential which is defined in clause 12.6.2.3 of ETSI TS 103 666-2 [10].
RQ1203_029	12.3.3.2	The bound SPB image download procedure shall use the following steps: 2) On reception of the "Si3.SetSpbmCredential" function command, the Secondary Platform Bundle Loader shall: a) Set the aSpbmCredential to IDS_CREDENTIAL_PARAMETER registry. b) Verify that the received SPB Manager's certificate for key agreement contained in the aSpbmCredential by using the certification path verification as defined in clause 12.2.1.1.4 of ETSI TS 103 666-2 [10]. c) Select the appropriate Secondary Platform Bundle Loader certificate that shall be verifiable by the trusted public key which is identified by the aSspPkIdForSpblVerification contained in the aSpbmCredential. d) Generate an ephemeral key pair and generate the first session key. e) Build the aSspCredential which is defined in clause 12.6.2.4 of ETSI TS 103 666-2 [10]. f) Set the aSspCredential to the TRE_CREDENTIAL_PARAMETER registry.
RQ1203_030	12.3.3.2	The bound SPB image download procedure shall use the following steps: 3) The Secondary Platform Bundle Loader shall return ANY_OK to the LBA.
RQ1203_031	12.3.3.2	The bound SPB image download procedure shall use the following steps: 4) The LBA shall call "Si3.GetSspCredential" function. The function command shall contain the index of TRE_CREDENTIAL_PARAMETER registry.
RQ1203_032	12.3.3.2	The bound SPB image download procedure shall use the following steps: 5) The Secondary Platform Bundle Loader shall return ANY_OK with the aSspCredential.
RQ1203_033	12.3.3.2	The bound SPB image download procedure shall use the following steps: 6) The LBA shall call the "Si2.GetBoundSpbImage" function. The function command shall contain aSspCredential, aTerminalInfo, and aRequestType.
RQ1203_034	12.3.3.2	The bound SPB image download procedure shall use the following steps: 7) On reception of the "Si2.GetboundSpbImage" function command, the SPB Manager shall: a) Generate the first session key. b) Decrypt the encrypted data contained in the aSspCredential by using the first session key. c) Verify the Secondary Platform Bundle Loader certificate by using the certification path verification as defined in clause 12.2.1.1.4 of ETSI TS 103 666-2 [10] with the public key identified by the aSspPkIdForSpblVerification. The aSspPkIdForSpblVerification shall be determined in the capability negotiation as defined in clause 12.3.3.1 of ETSI TS 103 666-2 [10]. d) Find the Secondary Platform Bundle identifier linked to the aCodeM contained in the aSspCredential.
RQ1203_035	12.3.3.2	If the aCodeM is a reference which is not linked to a Secondary Platform Bundle identifier, the steps 8 and 9 shall be performed. Otherwise the step 10 shall be performed after finishing the step 7: 8) The SPB Manager shall call the "Si1.HandleNotification" function. The function command shall contain the aSspInfoProtected, aTerminalInfo and aCodeM. 9) The service provider shall perform the Secondary Platform Bundle selection process as defined in clause 12.3.2.2 of ETSI TS 103 666-2 [10]. 10) The SPB Manager shall perform the eligibility check based on the aSspInfoProtected and the aTerminalInfo.

Req.ID	Clause	Description
RQ1203_036	12.3.3.2	If the aRequestType is "RequestSpbMetadata", the steps from 11 to 15 shall be performed. Otherwise, the step 16 shall be performed after finishing the step 10: 11) The SPB Manager shall build aSpbMetadata and link the aSpbMetadata to the aIdTransac contained in aSspCredential. 12) The SPB Manager shall return the aSpbMetadata to the LBA. 13) The LBA: a) Shall store the aSpbMetadata. b) May display the aSpbMetadata to the end user and require the end user intent if configured. 14) The LBA shall call the "Si2.GetBoundSpbImage" function. The function command shall containaSspCredential, aTerminalInfo and aBoundSpbImageByTransacId as aRequestType. 15) Upon reception of "Si2.GetBoundSpbImage" function command and if the received function command contains the aBoundImageByTransacId, the SPB Manager shall verify the step 7 of this procedure was performed with the same aSspCredential. 16) The SPB Manager shall: a) Generate TIME_STAMP and ephemeral key pair. b) Generate the second session key. c) Build an aBoundSpbImage as defined in clause 12.6.2.5 of ETSI TS 103 666-2 [10]. 17) The SPB Manager shall return the aBoundSpbImage to the LBA. 18) On reception of the "Si2.GetBoundSpbImage" response, the LBA shall verify that the aSpbMetdatanreceived in step 13 and the aSpbMetadata contained in the aBoundSpbImage are the same. If the LBA did not request aSpbMetadata previously, the LBA shall display the aSpbMetadata to the end user and request the end user intent if configured.
	12.3.4	***Installation procedure***
RQ1203_037	12.3.4	The installation procedure shall use the following steps: 1) The LBA shall call the "Si3.LoadBoundSpbInfo" function. The function command shall contain aDoOperateParameter contained in the aBoundSpbImage received from the SPB Manager.
RQ1203_038	12.3.4	The installation procedure shall use the following steps: 2) On reception of the "Si3.LoadBoundSpbInfo" function command, the Secondary Platform Bundle Loader shall: a) Verify SPBM certificate for digital signature. b) Verify the content in the aDoOperateParameter. c) Generate the second session key. d) Decrypt the aM-TimeStamp contained in the aDoOperateParameter by using the first session key and decrypt the aM-IMD and aM-ARP contained in the aDoOperateParameter by using the second session key. NOTE: The first session key and the second session key are generated during the download procedure as defined in clause 12.3.3 of ETSI TS 103 666-2 [10]. e) Verify the content in the decrypted data.
RQ1203_039	12.3.4	The installation procedure shall use the following steps: 3) The Secondary Platform Bundle Loader shall return ANY_OK to the LBA.
RQ1203_040	12.3.4	The installation procedure shall use the following steps: 4) The LBA shall perform the following steps for aNumberSegment times. The aNumberSegment shall be in the aBoundSpbImage: a) The LBA shall call the "Si3.LoadBoundSpbSds" function. The function command shall contain the aChangeSegmentParameter contained in the aBoundSpbImage. b) On reception of the "Si3.LoadBoundSpbSds" function command, the Secondary Platform Bundle Loader shall decrypt the aChangeSegmentParameter by using the second session key. If successful, the Secondary Platform Bundle Loader shall return ANY_OK to the LBA. c) The LBA shall call the "Si3.LoadBoundSpbSeg" function. The function command shall contain the aLoadSegmentParameter contained in the aBoundSpbImage. d) On reception of the "Si3.LoadBoundSpbSeg" function command, the Secondary Platform Bundle Loader shall decrypt the aLoadSegmentParameter by using the key obtained by decrypting the aChangeSegmentParameter. If successful, the Secondary Platform Bundle Loader shall return ANY_OK to the LBA.

<table>
<tr><th>Req.ID</th><th>Clause</th><th>Description</th></tr>
<tr><td></td><td>12.3.5</td><td>SSP activation code</td></tr>
<tr><td>RQ1203_041</td><td>12.3.5</td><td>The SSP activation code contains the information which is needed by the LBA to trigger the Secondary Platform Bundle provisioning procedure.
The SSP activation code is encoded using a URI, as defined in IETF RFC 3986 [18], using the following rules:
• The scheme shall have the value "lba".
• The authority shall have the value of the FQDN of the SPB Manager to which the terminal shall establish a connection to download a Secondary Platform Bundle.
• The path shall contain the action to be performed. The string "bundle" identifies the action to download a Secondary Platform Bundle.
• The query contains the other parameters, in the form of key/value pairs. The following keys are defined:
<table>
<tr><th>Key</th><th>Value</th><th>M/O/C</th></tr>
<tr><td>codem</td><td>It describes the CodeMatching identifier used to indicate the specific Secondary Platform Bundle which is linked to the CodeM during the bundle ordering process.</td><td>M</td></tr>
<tr><td>familyid</td><td>It describes the family identifier of the Secondary Platform Bundle.</td><td>O (see note)</td></tr>
<tr><td>oid</td><td>It indicates the OID of the custodian of the family identifier.</td><td>O</td></tr>
<tr><td colspan="3">NOTE: If oid is provided, familyid shall be present.</td></tr>
</table>
</td></tr>
<tr><td>RQ1203_042</td><td>12.3.5</td><td>The LBA shall reject SSP activation codes containing an unknown path value.</td></tr>
</table>

5.10.3 Secondary Platform Bundle management procedure

Reference: ETSI TS 103 666-2 [10], clause 12.4.

Req.ID	Clause	Description
	12.4.1	***Enable a Secondary Platform Bundle***
RQ1204_001	12.4.1	The procedure to enable a Secondary Platform Bundle installed on the iSSP shall use the following steps: 1) The end user selects the Secondary Platform Bundle to enable through the LBA (out of scope of the present document).
RQ1204_002	12.4.1	The procedure to enable a Secondary Platform Bundle installed on the iSSP shall use the following steps: 2) The LBA shall get the user intent if the Secondary Platform Bundle to enable is a Telecom Secondary Platform Bundle and if the user intent is configured in the Secondary Platform Bundle metadata.
RQ1204_003	12.4.1	The procedure to enable a Secondary Platform Bundle installed on the iSSP shall use the following steps: 3) The LBA shall send the Si3.EnableSpb command to the Secondary Platform Bundle Loader with the identifier of the Secondary Platform Bundle to enable.
RQ1204_004	12.4.1	The procedure to enable a Secondary Platform Bundle installed on the iSSP shall use the following steps: 4) If the Secondary Platform Bundle to enable is a Telecom Secondary Platform Bundle, the Secondary Platform Bundle Loader shall verify if it can be enabled as described in clause 12.6.5.5.7 of ETSI TS 103 666-2 [10].
RQ1204_005	12.4.1	The procedure to enable a Secondary Platform Bundle installed on the iSSP shall use the following steps: 5) The Secondary Platform Bundle Loader shall use the Si3.EnableSpb response to indicate the execution status of the command.
	12.4.2	***Disable a Secondary Platform Bundle***
RQ1204_006	12.4.2	The procedure to disable a Secondary Platform Bundle installed on the iSSP shall use the following steps: 1) The end user selects the Secondary Platform Bundle to disable through the LBA (out of scope of the present document).
RQ1204_007	12.4.2	The procedure to disable a Secondary Platform Bundle installed on the iSSP shall use the following steps: 2) The LBA shall get the user intent if the Secondary Platform Bundle to disable is a Telecom Secondary Platform Bundle and if the user intent is configured in the Secondary Platform Bundle metadata.

Req.ID	Clause	Description
RQ1204_008	12.4.2	The procedure to disable a Secondary Platform Bundle installed on the iSSP shall use the following steps: 3) The LBA shall send the Si3.DisableSpb command to the Secondary Platform Bundle Loader with the identifier of the Secondary Platform Bundle to disable.
RQ1204_009	12.4.2	The procedure to disable a Secondary Platform Bundle installed on the iSSP shall use the following steps: 4) The Secondary Platform Bundle Loader shall use the Si3.DisableSpb response to indicate the execution status of the command.
	12.4.3	***Delete a Secondary Platform Bundle***
RQ1204_010	12.4.3	The procedure to delete a Secondary Platform Bundle installed on the iSSP shall use the following steps: 1) The end user selects the Secondary Platform Bundle to delete through the LBA (out of scope of the present document).
RQ1204_011	12.4.3	The procedure to disable a Secondary Platform Bundle installed on the iSSP shall use the following steps: 2) The LBA shall get the user intent if the Secondary Platform Bundle to delete is a Telecom Secondary Platform Bundle and if the user intent is configured in the Secondary Platform Bundle metadata.
RQ1204_012	12.4.3	If the Secondary Platform Bundle to delete is currently disabled, steps 3 and 4 should be skipped: 3) The LBA shall disable the Secondary Platform Bundle by sending the Si3.DisableSpb command to the Secondary Platform Bundle Loader with the identifier of the Secondary Platform Bundle to disable.
RQ1204_013	12.4.3	4) The Secondary Platform Bundle Loader shall use the Si3.DisableSpb response to indicate the execution status of the command.
RQ1204_014	12.4.3	5) The LBA shall send the Si3.DeleteSpb command to the Secondary Platform Bundle Loader with the identifier of the Secondary Platform Bundle to delete.
RQ1204_015	12.4.3	The Secondary Platform Bundle Loader shall use the Si3.DeleteSpb response to indicate the execution status of the command.
	12.4.4	***SPB metadata retrieving procedure***
RQ1204_016	12.4.4	The SPB metadata retrieving procedure shall use the following steps: 1) The LBA shall call the "Si3.GetSpbMetadata" function. The function command shall contain the identifier of the Secondary Platform Bundle (aSpbId) corresponding to the SPB metadata that the LBA intends to retrieve.
RQ1204_017	12.4.4	The SPB metadata retrieving procedure shall use the following steps: 2) The Secondary Platform Bundle Loader shall extract the SPB metadata from the firmware session of the Secondary Platform Bundle container identified by the aSpbId contained in the "Si3.GetSpbMetadata" function command.
RQ1204_018	12.4.4	The SPB metadata retrieving procedure shall use the following steps: 3) The Secondary Platform Bundle Loader shall return ANY_OK with the SPB as the "Si3.GetSpbMetadata" function response.
	12.4.5	***SPB state retrieving procedure***
RQ1204_019	12.4.5	The SPB state retrieving procedure shall use the following steps: 1) The LBA shall call the "Si3.UpdateSpbState" function. The function command shall contain the index of SPB_ID registry and the Secondary Platform Bundle identifier (SpbId) corresponding to the SPB of which the LBA intends to retrieve the state.
RQ1204_020	12.4.5	The SPB state retrieving procedure shall use the following steps: 2) On reception of the "Si3.UpdateSpbState" function command, the Secondary Platform Bundle Loader shall: a) Set the SpbId to SPB_ID registry. b) Update the value of SPB_STATE registry with the current state of the Secondary Platform Bundle identified by the SpbId.
RQ1204_021	12.4.5	The SPB state retrieving procedure shall use the following steps: 3) The Secondary Platform Bundle Loader shall return ANY_OK to the LBA.
RQ1204_022	12.4.5	The SPB state retrieving procedure shall use the following steps: 4) The LBA shall call "Si3.GetSpbState" function. The function command shall contain the index of SPB_STATE registry.
RQ1204_023	12.4.5	The Secondary Platform Bundle Loader shall return ANY_OK with the value of SPB_STATE registry to the LBA.

5.10.4 Notification procedure

Reference: ETSI TS 103 666-2 [10], clause 12.5.

Req.ID	Clause	Description
	12.5.2	***Notification of the service provider***
RQ1205_001	12.5.2	If any of the following steps has to be notified to the service provider according to the configuration of the SPB metadata, the SPB Manager shall call the "Si1.HandleNotification" function after the execution of this step: • The eligibility check procedure, as defined in Annex C, has been executed. • The user rejected the download of a Secondary Platform Bundle. • The download of a bound Secondary Platform Bundle image. • The maximum retry attempts to download a Secondary Platform Bundle image has been reached. • The installation of a Secondary Platform Bundle. • The enablement, disablement or deletion of a Secondary Platform Bundle.
RQ1205_002	12.5.2	If, for this step, a notification containing a notification token was previously received from the LBA, as defined in clause 12.5.3 of ETSI TS 103 666-2 [10], the notification event contained in the "Si1.HandleNotification" function command shall be the notification event retrieved from this notification token.
	12.5.3	***Notification from the LBA***
RQ1205_003	12.5.3	The notification procedure consists of the following steps: 1) A Secondary Platform Bundle container is installed or the state of the Secondary Platform Bundle container is changed as defined in clauses 12.3 and 12.4 of ETSI TS 103 666-2 [10].
RQ1205_004	12.5.3	If the state of the Secondary Platform Bundle container shall be notified to the SPB Manager according to the configuration of the SPB metadata, the steps 2, 3, 4 and 7 shall be performed. 2) The LBA shall retrieve a notification token by using ANY_GET_PARAMETER with the index of OPERATION_TOKEN registry.
RQ1205_005	12.5.3	3) The LBA shall establish a TLS connection with the SPB Manager in server authentication mode.
RQ1205_006	12.5.3	4) The LBA shall call the "Si2.HandleNotification" function. The function command shall include a notification token.
RQ1205_007	12.5.3	The LBA shall store the notification token retrieved at step 2 until that notification token is successfully delivered to the SPB Manager. Once the notification is successfully delivered to the SPB Manager, the LBA shall delete that notification token.
RQ1205_008	12.5.3	Otherwise, the steps 5, 6 and 7 shall be performed. 5) The LBA shall establish a TLS connection with the SPB Manager in server authentication mode.
RQ1205_009	12.5.3	6) The LBA shall call the "Si2.HandleNotification" function without a notification token.
RQ1205_010	12.5.3	7) The SPB Manager shall respond to the LBA to notify a successful reception of the notification.

5.10.5 Interfaces and functions - Overview

Reference: ETSI TS 103 666-2 [10] clause 12.6.1.

Req.ID	Clause	Description
RQ1206_001	12.6.1	If the Service provider is the function requester and the Secondary Platform Bundle Manager is the Function provider the Si1 Interface provides the functions: • Si1.SelectSpb • Si1.CreateSPReference • Si1.FinalizePreparation • Si1.CancelPreparation
RQ1206_002	12.6.1	If the Secondary Platform Bundle Manager is the function requester and the Service provider is the Function provider the Si1 Interface provides the functions: • Si1.HandleNotification
RQ1206_003	12.6.1	If the Local Bundle Assistant is the function requester and the Secondary Platform Bundle Manager is the Function provider the Si2 Interface provides the functions: • Si2.GetSpbmCertificate • Si2.GetBoundSpbImage
RQ1206_004	12.6.1	If the Local Bundle Assistant is the function requester and the Secondary Platform Bundle Loader is the Function provider the Si3 Interface provides the functions: • Si3.GetSspInfo • Si3.SetSpbmCredential • Si3.LoadBoundSpbInfo • Si3.LoadBoundSpbSds • Si3.LoadBoundSpbSeg • Si3.GetSspCredential • Si3.EnableSpb • Si3.DisableSpb • Si3.DeleteSpb

5.10.6 Interfaces and functions - Common features

Reference: ETSI TS 103 666-2 [10] clause 12.6.2.

Req.ID	Clause	Description
RQ1206_005	12.6.2.2.1	The SSP information comprises SspInfoPublic and sspInfoProtected. The SPB Manager shall perform eligibility check based on Annex C of ETSI TS 103 666-2 [10] by using the received SspInfoPublic and sspInfoProtected.
RQ1206_006	12.6.2.2.2	The SspInfoPublic is used during the capability negotiation procedure defined in clause 12.3.3 of ETSI TS 103 666-2 [10] to provide the SPB Manager with the trusted public key identifiers and cryptographic algorithms supported by the Secondary Platform Bundle Loader allowing the SPB Manager to select an appropriate certificate and cryptographic algorithm.
RQ1206_007	12.6.2.2.2	aSpblSpecVerInfo: • the release of the specification that is implemented by the Secondary Platform Bundle Loader. The first byte indicates the major version of the specification. The second byte indicates the minor version of the specification.
RQ1206_008	12.6.2.2.2	aSspPkIdListForSpbmVerification: • For aSspGeneralCryptoInfo, the list indicates the Public Key identifiers supported by the Secondary Platform Bundle Loader that allows for the Secondary Platform Bundle Loader to verify the SPBM certificate chain. • For aSspFamilyCryptoInfo, this list indicates the Public Key identifiers only allowed for the loading Secondary Platform Bundles with the aSpbFamilyId contained in the same SspFamilyCryptoInfoBlock. • For aSspOidCryptoInfo, this list indicates the Public Key identifiers only allowed for loading Secondary Platform Bundles with the aSpbFamilyId contained in the same SspFamilyCryptoInfo and the OID contained in the same SspOidCryptoInfoBlock.
RQ1206_009	12.6.2.2.2	aSspPkIdListForSpblVerification: • For aSspGeneralCryptoInfo, the list indicates the Public Key identifiers supported by the Secondary Platform Bundle Loader that allows for the SPB Manager to verify the SPBL certificate chain.

Req.ID	Clause	Description
		• For aSspFamilyCryptoInfo, the list indicates the Public Key identifiers only allowed for the loading of Secondary Platform Bundles with the aSpbFamilyId contained in the same SspFamilyCryptoInfoBlock. • For aSspOidCryptoInfo, the list indicates the Public Key identifiers only allowed for the loading of Secondary Platform Bundles with the aSpbFamilyId contained in the same SspFamilyCryptoInfo and the OID contained in the same SspOidCryptoInfoBlock.
RQ1206_010	12.6.2.2.2	aKeyAgreementAlgorithmList: • For aSspGeneralCryptoInfo, the list indicates the algorithm identifiers for key agreement algorithms supported by the Secondary Platform Bundle Loader. • For aSspFamilyCryptoInfo, the list indicates the key agreement algorithms only allowed for the loading of Secondary Platform Bundles with the aSpbFamilyId contained in the same SspFamilyCryptoInfoBlock. • For aSspOidCryptoInfo, the list indicates the key agreement algorithms only allowed for the loading of Secondary Platform Bundles with the aSpbFamilyId in the same SspFamilyCryptoInfoBlock and the OID contained in the same SspOidCryptoInfoBlock.
RQ1206_011	12.6.2.2.2	aCipherAlgorithmList: • For aSspGeneralCryptoInfo, the list indicates the algorithm identifiers of data encryption algorithms supported by the Secondary Platform Bundle Loader. • For aSspFamilyCryptoInfo, the list indicates the data encryption algorithms only allowed for the loading of Secondary Platform Bundles with the aSpbFamilyId contained in the same SspFamilyCryptoInfoBlock. • For aSspOidCryptoInfo, the list indicates the data encryption algorithms only allowed for the loading of Secondary Platform Bundles with the aSpbFamilyId in the same aSspFamilyCryptoInfoBlock and the OID contained in the same aSspOidCryptoInfoBlock.
RQ1206_012	12.6.2.2.2	aSpbFamilyId: • a family identifier supported by the Secondary Platform Bundle Loader.
RQ1206_013	12.6.2.2.2	aOid: • the OID of a custodian of the family identifier aSpbFamilyId.
RQ1206_014	12.6.2.2.3	The Secondary Platform Bundle Loader shall provide the SPB Manager with sspInfoProtected containing the primary platform identifier and family identifier-specific SSP information.
RQ1206_015	12.6.2.2.3	aPpIdentifier: • the Primary Platform identifier as defined in clause 7.5 of ETSI TS 103 666-2 [10].
RQ1206_016	12.6.2.2.3	aPartNumberId: • the Part Number identifier as defined in clause 7.7 of ETSI TS 103 666-2 [10] (the identifier of the Part Number in the format of UUID as defined in GlobalPlatform Open Firmware Loader for Tamper Resistant Element [13].) The aPartNumberId shall be used by the SPB Manager to identify the Primary Platform manufacturer and the model of the Primary Platform.
RQ1206_017	12.6.2.2.3	aMaxSpbSizeSupported: • it indicates the maximum size, in bytes, of the Secondary Platform Bundle container that the iSSP supports. The value of the aMaxSpbSizeSupported shall be the same as the value of MK_MEMORY_PARTITION_SIZE as defined in clause 7.2 of GlobalPlatform VPP - Concepts and Interfaces [15].
RQ1206_018	12.6.2.2.3	aFamilySpecificSspInfo: • it shall include the family identifier-specific SSP information which may be defined for that family identifier.
RQ1206_019	12.6.2.2.3	aSpbFamilyId: • the family identifier of the Secondary Platform Bundle.
RQ1206_020	12.6.2.2.3	aOidSpecificSspInfoBlock: • it shall include the family identifier-specific SSP information which may be defined by an organization that is responsible for that family identifier and referenced by aOID.
RQ1206_021	12.6.2.3	The SPBM credential shall be delivered to the Secondary Platform Bundle Loader during the Secondary Platform Bundle provisioning procedure in clause 12.3 of ETSI TS 103 666-2 [10].
RQ1206_022	12.6.2.3	The LBA shall provide the SPBM credential to the Secondary Platform Bundle Loader by calling the "Si3.GetSspCredential" function and obtain SSP credential as the response.
RQ1206_023	12.6.2.3	aCodeM: • the value of the CODE_M as defined in GlobalPlatform Open Firmware Loader for Tamper Resistant Element [13]. It indicates the code matching identifier for a Secondary Platform Bundle image within a SPB Manager.

Req.ID	Clause	Description
RQ1206_024	12.6.2.3	aChallengeS: • the value of CHALLENGE_S as defined in GlobalPlatform Open Firmware Loader for Tamper Resistant Element [13]. The aChallengeS is generated by the SPB Manager and used in authentication of the Secondary Platform Bundle Loader.
RQ1206_025	12.6.2.3	aSpbFamilyId: • the family identifier of the Secondary Platform Bundle.
RQ1206_026	12.6.2.3	aCustodianOid: • the OID of a custodian of the aSpbFamilyId. The custodian shall be associated with a certification path.
RQ1206_027	12.6.2.3	aSpbmKaCertificates: • SPBM Certificates for key agreement.
RQ1206_028	12.6.2.3	aSspCiPkIdToBeUsed: • CI Public Key identifier for SPBL Certificate which shall be used by the Secondary Platform Bundle Loader for signature generation.
RQ1206_029	12.6.2.3	aSspCryptoToBeUsed: • Algorithm identifiers for data encryption which shall be used by the Secondary Platform Bundle Loader and the SPB Manager.
RQ1206_030	12.6.2.4	The SSP credential is delivered from the Secondary Platform Bundle Loader to the SPB Manager for authentication, key agreement, and for the binding of a Secondary Platform Bundle container.
RQ1206_031	12.6.2.4	aTbsSspImageSessionToken: it contains: • aIdTransac: the ID_TRANSAC as defined in GlobalPlatform Open Firmware Loader for Tamper Resistant Element [13]. • aEPkSpblKa: the SPBL ephemeral public key. • aSpbmKaPkIdToBeUsed: the subject key identifier of the SPBM certificate for key agreement which shall be used to generated the first session key. • aUuidL: the UUIDL as defined in GlobalPlatform Open Firmware Loader for Tamper Resistant Element [13].
RQ1206_032	12.6.2.4	aSspImageSessionTokenSignature: • the signature of aTbsSspImageSessionToken which can be verified by SPBL Certificate.
RQ1206_033	12.6.2.4	aM-SSP: • EncryptedBlock of data containing SPBL Certificate, TbsSspToken, and SSP Token signature. The SSP Token, SSP Token signature, and aM-SSP are generated by the Secondary Platform Bundle Loader as defined in clause 12.6.5.5.2 of ETSI TS 103 666-2 [10].
RQ1206_034	12.6.2.4	aEncryptionType: • it indicates the encryption algorithm used to generate the items of type EncryptedBlock.
RQ1206_035	12.6.2.4	EncryptedBlock: • data structure containing the encrypted message and the integrity check.
RQ1206_036	12.6.2.4	aSpblCertChain: • it contains Certificates used for the SPB Manager to verify SPBL Certificate.
RQ1206_037	12.6.2.4	aTbsSspToken: it contains: • aCodeM: the value of the CODE_M as defined in GlobalPlatform Open Firmware Loader for Tamper Resistant Element [13]. • aChallengeS: the value of CHALLENGE_S as defined in GlobalPlatform Open Firmware Loader for Tamper Resistant Element [13]. • aSspInfoProtected: the protected SSP information.
RQ1206_038	12.6.2.5	The Secondary Platform Bundle container shall be bound to the iSSP as defined in clause 12.6.4.3 of ETSI TS 103 666-2 [10] and delivered to the LBA as the bound SPB image.
RQ1206_039	12.6.2.5	aImageOwnerId: • Owner Identifier of the Secondary Platform Bundle container.
RQ1206_040	12.6.2.5	aNumberSegment: • Number of Segment Structures in the bound SPB image.
RQ1206_041	12.6.2.5	aServerNotifyBaseUrls: • URLs of the servers for notifications.
RQ1206_042	12.6.2.5	aImageMakerId: • Identifier of the Secondary Platform Bundle maker as defined in GlobalPlatform Open Firmware Loader for Tamper Resistant Element [13].
RQ1206_043	12.6.2.5	aMetaDataImage: • Metadata of the image from the Image Maker.

Req.ID	Clause	Description
RQ1206_044	12.6.2.5	aM-IMD: • EncryptedBlock of Image Descriptor as defined in GlobalPlatform Open Firmware Loader for Tamper Resistant Element [13].
RQ1206_045	12.6.2.5	aM-ARP: • EncryptedBlock of ATK.ARP.ECDSA as defined in GlobalPlatform Open Firmware Loader for Tamper Resistant Element [13].
RQ1206_046	12.6.2.5	aM-TimeStamp: • EncrytedBlock of TIME_STAMP as defined in GlobalPlatform Open Firmware Loader for Tamper Resistant Element [13].
RQ1206_047	12.6.2.5	aSpbmToken: • Data structure containing TbsSpbmToken and signature of the TbsSpbmToken.
RQ1206_048	12.6.2.5	aTbsSpbmToken: • Data structure containing the ephemeral public key of the SPB Manager and the image session identifier (ID_TRANSAC).
RQ1206_049	12.6.2.5	aSpbmCerts: • List of the Certificate of the certification path from a trusted certificate to the SPBM certificate.
RQ1206_050	12.6.2.5	aDoOperateParameter: • The parameter for the OFL_DO_OPERATE command as defined in GlobalPlatform Open Firmware Loader for Tamper Resistant Element [13] including the SPB metadata.
RQ1206_051	12.6.2.5	aChangeSegmentParameter: • The parameter for the OFL_CHANGE_SEGMENT command as defined in GlobalPlatform Open Firmware Loader for Tamper Resistant Element [13].
RQ1206_052	12.6.2.5	aLoadSegmentParameter: • The parameter for the OFL_LOAD_SEGMENT command as defined in GlobalPlatform Open Firmware Loader for Tamper Resistant Element [13].
RQ1206_053	12.6.2.6	The SPB metadata contains specific information of the Secondary Platform Bundle. During the download procedure described in clause 12.3.3 of ETSI TS 103 666-2 [10], the SPB metadata shall be provided to the LBA in a plaintext.
RQ1206_054	12.6.2.6	The SPB metadata contains specific information of the Secondary Platform Bundle. After the Secondary Platform Bundle is successfully installed, the SPB metadata shall be accessible by the LBA via Si3 interface irrespective of the state of that Secondary Platform Bundle.
RQ1206_055	12.6.2.6	The SPB metadata shall include the following: • aSpbId: identifier of the Secondary Platform Bundle. • aSpbFamilyId: family identifier of the Secondary Platform Bundle as defined in clause 9.7 of ETSI TS 103 666-2 [10]. • aCustodianOid: OID of one of the custodians associated with a SpbFamilyId which defines specific requirement (e.g. trusted CIs, product certification, operational modes of the terminal) applied to this Secondary Platform Bundle.
RQ1206_056	12.6.2.6	The SPB metadata may include the following: • aSupportedCustodianList: list of OIDs of custodians associated with supported certification path used to load the Secondary Platform Bundle. If the aSupportedCustodianList contains multiple OIDs, the first OBJECT IDENTIFIER denotes the most preferred custodian to select a certification path. The aSupportedCustodianList may contain the aCustodianOid. • aSpbNotificationConfig: it includes the configuration set by a service provider as per a notification recipient. The configuration shall include the address of a notification recipient and the list of events which shall be notified. NOTE: Eligibility check and maximum retry attempts notification status are only intended to be used for Si1 notifications. aFamilySpecificData: family identifier-specific metadata defined as per the family identifier. • aOidSpecificData: family identifier-specific metadata defined by custodian(s) of the family identifier. The aOidSpecificData may consists of multiple OidSpecificInfoBlock data structures. Each OidSpecificInfoBlock shall have a custodian-defined metadata and the OID identifying that custodian-defined metadata.
RQ1206_057	12.6.2.7	The terminal information contains details about the capabilities of the terminal. It is delivered by the LBA to the SPB Manager.
RQ1206_058	12.6.2.7	aLbaSpecVerInfo: • the release of the specification that is implemented by the LBA. The first byte indicates the major version of the specification. The second byte indicates the minor version of the specification.

Req.ID	Clause	Description
RQ1206_059	12.6.2.7	aSpbFamilyId: • a family identifier of the Secondary Platform Bundle.
RQ1206_060	12.6.2.7	aFamilySpecificTerminalInfoBlock: • it shall include the family identifier-specific terminal information which may be defined for that family identifier.
RQ1206_061	12.6.2.7	aOidSpecificTerminalInfoBlock: • it shall include family identifier-specific terminal information which may be defined by an organization that is responsible for that family identifier and referenced by the OID.
RQ1206_062	12.6.2.8	The notification token contains the information about the state change of the Secondary Platform Bundle container.
RQ1206_063	12.6.2.8	The Secondary Platform Bundle Loader shall generate the notification token after installation, enabling, disabling or deleting of the Secondary Platform Bundle container if it is configured in the SPB metadata.
RQ1206_064	12.6.2.8	Prior to the generation of the notification token, the notification counter of the firmware session of the Secondary Platform Bundle container as described in clause 7.3.1.5 of ETSI TS 103 666-2 [10] shall be incremented by one.
RQ1206_065	12.6.2.8	If the maximum value is reached, the counter shall return to the initial value.
RQ1206_066	12.6.2.8	After generating the notification token, the Secondary Platform Bundle Loader shall set the value of the notification token data object into OPERATION_TOKEN registry.
RQ1206_067	12.6.2.8	The LBA shall use ANY_GET_PARAMETER command with the index of OPERATION_TOKEN registry to retrieve the most recently generated notification token.
RQ1206_068	12.6.2.8	aSpbId: • the identifier of the Secondary Platform Bundle. The aSpbId shall be the Public image UUID.
RQ1206_069	12.6.2.8	aNotificationEvent: • it indicates the procedure related to this notification.
RQ1206_070	12.6.2.8	aCounter: • the notification counter value managed in the firmware session of the Secondary Platform Bundle identified by the aSpbId.
RQ1206_071	12.6.2.8	aNotificationTokenHash: • the hashed value generated by an HMAC-SHA-256 as defined in IETF RFC 4868 [20] of the message being the string concatenating the aSpbId, the aNotificationEvent, the aCounter, and the Primary Platform identifier (of 32 bytes, as defined in clause 7.5 of ETSI TS 103 666-2 [10]), and with the secondary platform private identifier described in clause 9.4.5 of ETSI TS 103 666-2 [10] as the key. aNotificationTokenHash is therefore computed with following parameters: aNotificationTokenHash = HMAC-SHA-256 (message, key) where: message = aSpbId | aNotificationEvent | aCounter | aPpId, and key = aPrivateSpbId. NOTE: The Primary Platform Identifier is used to compute the aNotificationTokenHash but is not included in the aTbhNotificationToken.

5.10.7 Interfaces and functions - Si1 interface

Reference: ETSI TS 103 666-2 [10] clause 12.6.3.

Req.ID	Clause	Description
	12.6.3.1	***Overview***
RQ1206_072	12.6.3.1	The Si1 interface is used between the service provider and the SPB Manager to prepare the download of a Secondary Platform Bundle.
RQ1206_073	12.6.3.1	The binding of the Si1 interface shall be performed over Hypertext Transfer Protocol version 2 (HTTP/2) as defined in IETF RFC 7540 [26] and the Transport Layer Security (TLS) version 1.3 or higher in mutual authentication mode as defined in IETF RFC 8446 [27].
RQ1206_074	12.6.3.1	The service provider shall be in charge of managing the connection establishment to the SPB Manager for the Si1 interface.
RQ1206_075	12.6.3.1	The service provider shall use HTTP POST request message with HTTP path 'etsi/issp/si1/asn1' to deliver any function command over the Si1 interface.

Req.ID	Clause	Description
	12.6.3.2	***Si1 common headers***
RQ1206_076	12.6.3.2.1	aFunctionRequesterId: • identifier of the function requester.
RQ1206_077	12.6.3.2.1	aFunctionCallId: • identifier of the function call. This identifier is used to manage function call retries.
RQ1206_078	12.6.3.2.2	aFunctionExecutionStatus: • indicates the status after the execution of the function.
	12.6.3.3	***Si1 error codes***
RQ1206_079	12.6.3.3	For error codes used to indicate an error over the Si1 interface, Table 12.5 of ETSI TS 103 666-2 [10] gives the applicability matrix according to the Si1 function.
	12.6.3.4	***Si1.SelectSpb***
RQ1206_080	12.6.3.4.1	The "Si1.SelectSpb" function shall be used by the service provider during the Secondary Platform Bundle selection as defined in clause 12.3.2.2 of ETSI TS 103 666-2 [10].
RQ1206_081	12.6.3.4.1	The service provider shall use the "Si1.SelectSpbm" function to select a Secondary Platform Bundle that matches the terminal and the SSP capabilities.
RQ1206_082	12.6.3.4.1	The body part of the HTTP POST request for the "Si1.SelectSpbm" function command shall contain Si1SelectSpbCommand defined as follows: • aSi1CommandHeader: – header of the command as defined in clause 12.6.3.2.1 of ETSI TS 103 666-2 [10]. It may be used by aNotificationReceiverId in subsequent Si1.HandleNotification calls related to this selection. • aSpbId: – identifier of the Secondary Platform Bundle to reserve. • aSpbType: – type of Secondary Platform Bundle in which the SPB Manager shall select an available Secondary Platform Bundle identifier. • aPpIdentifier: – the primary platform identifier to link with the Secondary Platform Bundle reserved by the Si1.SelectSpb function. • aCodeM: – CodeM to be linked with the Secondary Platform Bundle reserved by the Si1.SelectSpb function. This parameter shall be present if the creation of the service provider reference defined in clause 12.3.2.3 of ETSI TS 103 666-2 [10] has been previously executed. • aFlagFinalize: – Boolean that indicates whether the "Si1.FinalizePreparation" function will be called later. • aCustodianSpecificInfoBlock: – specific parameter which may be defined by the custodian of the family identifier issuing the command. How this parameter is handled by the SPB Manager is out of scope of the present document. • aServiceProviderSpecificInfoBlock: – specific parameter which may be defined by the service provider. How this parameter is handled by the SPB Manager is out of scope of the present document.
RQ1206_083	12.6.3.4.2	Upon reception of the "Si1.SelectSpb" function command, the SPB Manager shall: • Store the value of aSi1CommandHeader.
RQ1206_084	12.6.3.4.2	Upon reception of the "Si1.SelectSpb" function command, the SPB Manager shall: • If the Secondary Platform Bundle identifier (spbId) was provided as input data: – return an error with the code eSpbIdNotAvailable if the spbId is not available; – return an error with the code eSpbIdUnknown if the spbId does not exist; – return an error with the code eSpbTypeMismatch if a Secondary Platform Bundle type (spbType) was provided as input data and does not match the type of the spbId.
RQ1206_085	12.6.3.4.2	Upon reception of the "Si1.SelectSpb" function command, the SPB Manager shall: • If a Secondary Platform Bundle type (spbType) was provided as input data: – return an error with the code eSpbTypeUnknown if the spbType is unknown to the SPB Manager; – return an error with the code eSpbTypeNotAvailable if the SPM Manager cannot find an available spbId corresponding to the requested sbpType.
RQ1206_086	12.6.3.4.2	Upon reception of the "Si1.SelectSpb" function command, the SPB Manager shall: • Return an error with the code eCodeMNotAllowed if a CodeM was provided as input data and is already linked to another Secondary Platform Bundle identifier.

Req.ID	Clause	Description
RQ1206_087	12.6.3.4.2	Upon reception of the "Si1.SelectSpb" function command, the SPB Manager shall: • Store the CodeM if provided as input data and is not known to the SPB Manager.
RQ1206_088	12.6.3.4.2	Upon reception of the "Si1.SelectSpb" function command, the SPB Manager shall: • Reserve a Secondary Platform Bundle among those available in its inventory and that corresponds to the requested spbId and/or sbpType.
RQ1206_089	12.6.3.4.2	Upon reception of the "Si1.SelectSpb" function command, the SPB Manager shall: • Link the CodeM with the reserved spbId.
RQ1206_090	12.6.3.4.2	Upon reception of the "Si1.SelectSpb" function command, the SPB Manager shall: • Link the reserved spbId to the primary platform identifier if the function command contains the primary platform identifier (aPpIdentifier).
RQ1206_091	12.6.3.4.2	Upon reception of the "Si1.SelectSpb" function command, the SPB Manager shall: • Memorize whether the "Si1.FinalizePreparation" function will be called later. If aFlagFinalize is not present, it is considered as set to FALSE.
RQ1206_092	12.6.3.4.2	Upon reception of the "Si1.SelectSpb" function command, the SPB Manager shall: • Build Si1SelectSpbResponse containing either an error code if one of the above steps has failed or the selected Secondary Platform identifier (spbId) together with its type (sbpType) and, optionally the Primary Platform Identifier (aPpIdentifier) if it was provided in the command and the CodeM (aCodeM) if it was provided in the incoming command. Si1SelectSpbResponse may also contain family identifier and/or service provider specific information. Their content is not in the scope of the present document.
RQ1206_093	12.6.3.4.2	Upon reception of the "Si1.SelectSpb" function command, the SPB Manager shall: • Send the response to the service provider.
RQ1206_094	12.6.3.4.3	The body part of the HTTP POST response for the "Si1.SelectSpb" function shall contain Si1SelectSpbResponse defined as follows: • aSi1ResponseHeader: header of the response as defined in clause 12.6.3.2.2 of ETSI TS 103 666-2 [10]. • aSpbId:identifier of the Secondary Platform Bundle reserved by the SPB Manager. • aSpbType: type of Secondary Platform Bundle tied to aSpbId. • aPpIdentifier: identifier of the primary platform linked with aSpbId, if present in the incoming command. • aCodeM: CodeM to linked with the aSpbId, if present in the incoming command. • aCustodianSpecificInfoBlock: specific parameter which may be defined by the custodian of the family identifier issuing the response. How this parameter is handled by the SPB Manager is out of scope of the present document. • aServiceProviderSpecificInfoBlock: specific parameter which may be defined by the service provider. How this parameter is handled by the SPB Manager is out of scope of the present document. • aFamilySpecificSelectSpbmResponse: family identifier-specific parameter which may be defined for that family identifier. How this parameter is handled by the SPB Manager is out of scope of the present document.
	12.6.3.5	***Si1.CreateSPReference***
RQ1206_095	12.6.3.5.1	The "Si1.CreateSPReference" function shall be used by the service provider during the procedure of creation of a service provider reference as defined in clause 12.3.2.3. of ETSI TS 103 666-2 [10].
RQ1206_096	12.6.3.5.1	The service provider may use the "Si1.CreateSPReference" function to create a reference shared between the service provider and the SPB Manager. This reference, i.e. CodeM shall be provided to the End User by the service provider as part of the activation code, allowing the End User to trigger the download procedure as defined in clause 12.3.3 of ETSI TS 103 666-2 [10].
RQ1206_097	12.6.3.5.1	The body part of the HTTP POST request for the "Si1.CreateSPReference" function command shall contain Si1.CreateSPReferenceCommand defined as follows: • aSi1CommandHeader: header of the command as defined in clause 12.6.3.2.1 of ETSI TS 103 666-2 [10]. It may be used by aNotificationReceiverId in subsequent Si1.HandleNotification calls related to the CodeM provided as input parameter or generated by the SPB Manager. • aSpbId: identifier of the Secondary Platform Bundle. This parameter shall be present if the Secondary Platform Bundle selection procedure has been executed first, else it shall be ignored. • aCodeM: CodeM generated by the service provider. • aTaskType: type of task associated with the reference. • aCustodianSpecificInfoBlock: specific parameter which may be defined by the custodian of the family identifier issuing the command. How this parameter is handled by the SPB Manager is out of scope of the present document.

Req.ID	Clause	Description
		• aServiceProviderSpecificInfoBlock: specific parameter which may be defined by the service provider. How this parameter is handled by the SPB Manager is out of scope of the present document.
RQ1206_098	12.6.3.5.2	Upon reception of the "Si1.CreateSPReference" function command, the SPB Manager shall: • Store the value of aSi1CommandHeader.
RQ1206_099	12.6.3.5.2	Upon reception of the "Si1.CreateSPReference" function command, the SPB Manager shall: • Return an error with the code eTaskTypeUnknown if the Si1TaskType is not eSi1TaskType-DownloadSPB.
RQ1206_100	12.6.3.5.2	Upon reception of the "Si1.CreateSPReference" function command, the SPB Manager shall: • Return an error with the code eTaskNotAllowed if the function caller is not allowed to use the Si1TaskType. NOTE: How the function caller is allowed to use is not in the scope of ETSI TS 103 666-2 [10].
RQ1206_101	12.6.3.5.2	Upon reception of the "Si1.CreateSPReference" function command, the SPB Manager shall: • Generate a CodeM if it was not provided as input data and ensure that it is unique on its own context.
RQ1206_102	12.6.3.5.2	Upon reception of the "Si1.CreateSPReference" function command, the SPB Manager shall: • Return an error with the code eCodeMNotAllowed if the CodeM was provided as input data and is already linked to another Secondary Platform Bundle identifier.
RQ1206_103	12.6.3.5.2	Upon reception of the "Si1.CreateSPReference" function command, the SPB Manager shall: • Store the CodeM.
RQ1206_104	12.6.3.5.2	Upon reception of the "Si1.CreateSPReference" function command, the SPB Manager shall: • Store the value of aSi1CommandHeader.
RQ1206_105	12.6.3.5.2	Upon reception of the "Si1.CreateSPReference" function command, the SPB Manager shall: • Build Si1CreateSPReferenceResponse containing either an error code if one of the above step has failed or the CodeM (aCodeM) provided as input data or generated by the SPB Manager and the Secondary Platform identifier (spbId) if it was provided as input data. Si1CreateSPReferenceResponse may also contain family identifier and/or service provider specific information. Their content is not in the scope of the present document.
RQ1206_106	12.6.3.5.2	Upon reception of the "Si1.CreateSPReference" function command, the SPB Manager shall: • Send the response to the service provider.
RQ1206_107	12.6.3.5.3	The body part of the HTTP POST response for the "Si1.CreateSPReference" function shall contain Si1CreateSPReferenceResponse defined as follows: • aSi1ResponseHeader: header of the response as defined in clause 12.6.3.2.2 of ETSI TS 103 666-2 [10]. • aCodeM: CodeM generated by the SPB manager if not present in the incoming command or CodeM as it was in the incoming command. • aSpbId: identifier of the Secondary Platform Bundle as if was in the incoming command. • aCustodianSpecificInfoBlock: specific parameter which may be defined by the custodian of the family identifier issuing the response. How this parameter is handled by the SPB Manager is out of scope of the present document. • aServiceProviderSpecificInfoBlock: specific parameter which may be defined by the service provider. How this parameter is handled by the SPB Manager is out of scope of the present document.
	12.6.3.6	***Si1.FinalizePreparation***
RQ1206_108	12.6.3.6.1	The "Si1.CreateSPReference" function shall be used by the service provider during the procedure of creation of a service provider reference as defined in clause 12.3.2.3. of ETSI TS 103 666-2 [10].
RQ1206_109	12.6.3.6.1	If the selection of the Secondary Platform Bundle procedure, as defined in clause 12.3.2.2 of ETSI TS 103 666-2 [10] has been executed after the creation of the CodeM procedure, as defined in clause 12.3.2.3 of ETSI TS 103 666-2 [10], the service provider may use the "Si1.FinalizePreparation" function to indicate that its internal procedures are completed, e.g. the provisioning of its technical platforms or data bases.

Req.ID	Clause	Description
RQ1206_110	12.6.3.6.1	If the service provider has set aFlagFinalize to TRUE in the "Si1.SelectSpb" function command, the SPB Manager shall wait for the completion of the Secondary Platform Bundle selection process as described in clause 12.3.2.2 of ETSI TS 103 666-2 [10] (i.e. after it has sent the response to the "Si1.FinalizePreparation" function related to this Secondary Platform Bundle) to continue with the Bound SPB image download as defined in clause 12.3.3.2. of ETSI TS 103 666-2 [10].
RQ1206_111	12.6.3.6.1	The body part of the HTTP POST request for the "Si1.FinalizePreparation" function command shall contain Si1.FinalizePreparationCommand defined as follows: • aSi1CommandHeader: header of the command as defined in clause 12.6.3.2.1 of ETSI TS 103 666-2 [10]. It may be used by aNotificationReceiverId in subsequent Si1.HandleNotification calls related to aCodeM. • aCodeM: reference to the preparing procedure to finalize. • aSpbId: identifier of the Secondary Platform Bundle as if was in the incoming command. • aCustodianSpecificInfoBlock: specific parameter which may be defined by the custodian of the family identifier issuing the command. How this parameter is handled by the SPB Manager is out of scope of the present document. • aSrviceProviderSpecificInfoBlock: specific parameter which may be defined by the service provider. How this parameter is handled by the SPB Manager is out of scope of the present document.
RQ1206_112	12.6.3.6.2	Upon reception of the "Si1.FinalizePreparation" function command, the SPB Manager shall: • Store the value of aSi1CommandHeader.
RQ1206_113	12.6.3.6.2	Upon reception of the "Si1.FinalizePreparation" function command, the SPB Manager shall: • Verify the CodeM provided as input data.
RQ1206_114	12.6.3.6.2	Upon reception of the "Si1.FinalizePreparation" function command, the SPB Manager shall: • Return an error with the code eCodeMUnknown if the CodeM is unknown to the SPB Manager.
RQ1206_115	12.6.3.6.2	Upon reception of the "Si1.FinalizePreparation" function command, the SPB Manager shall: • Return an error with the code eCodeMNotAllowed if the CodeM is not linked to a Secondary Platform Bundle identifier.
RQ1206_116	12.6.3.6.2	Upon reception of the "Si1.FinalizePreparation" function command, the SPB Manager shall: • Build Si1FinalizePreparationResponse containing either an error code if the above step has failed or the CodeM (aCodeM) provided as input data. Si1FinalizePreparationResponse may also contain family identifier and/or service provider specific information. Their content is not in the scope of the present document.
RQ1206_117	12.6.3.6.2	Upon reception of the "Si1.FinalizePreparation" function command, the SPB Manager shall: • Send the response to the service provider.
RQ1206_118	12.6.3.6.2	Upon reception of the "Si1.CreateSPReference" function command, the SPB Manager shall: • Allow the bound SPB image download procedure as defined in clause 12.3.3.2 of ETSI TS 103 666-2 [10].
RQ1206_119	12.6.3.6.3	The body part of the HTTP POST response for the "Si1.finalizePreparation" function shall contain Si1FinalizePreparationResponse defined as follows: • aSi1ResponseHeader: header of the response as defined in clause 12.6.3.2.2 of ETSI TS 103 666-2 [10]. • aCodeM: CodeM as it was in the incoming command. • aCustodianSpecificInfoBlock: specific parameter which may be defined by the custodian of the family identifier issuing the response. How this parameter is handled by the SPB Manager is out of scope of the present document. • aServiceProviderSpecificInfoBlock: specific parameter which may be defined by the service provider. How this parameter is handled by the SPB Manager is out of scope of the present document.
	12.6.3.7	***Si1.CancelPreparation***
RQ1206_120	12.6.3.7.1	The "Si1.CancelPreparation" function shall be used by the service provider to cancel a pending preparation procedure as defined in clause 12.3.2 of ETSI TS 103 666-2 [10].

Req.ID	Clause	Description
RQ1206_121	12.6.3.7.1	The body part of the HTTP POST request for the "Si1.CancelPreparation" function command shall contain Si1CancelPreparationCommand defined as follows: • aSi1CommandHeader: header of the command as defined in clause 12.6.3.2.1 of ETSI TS 103 666-2 [10]. It may be used by aNotificationReceiverId in subsequent Si1.HandleNotification calls related to aCodeM or aSpbId. • aCodeM: task's reference to cancel. This parameter shall be present if aSpbId is not provided as input parameters. • aSpbId: identifier of the Secondary Platform Bundle associated to the procedure to cancel. This parameter shall be present if aCodeM is not provided as input parameters. • aCustodianSpecificInfoBlock: specific parameter which may be defined by the custodian of the family identifier issuing the command. How this parameter is handled by the SPB Manager is out of scope of the present document. • aServiceProviderSpecificInfoBlock: specific parameter which may be defined by the service provider. How this parameter is handled by the SPB Manager is out of scope of the present document.
RQ1206_122	12.6.3.7.2	Upon reception of the "Si1.CancelPreparation" function command, the SPB Manager shall: • Store the value of aSi1CommandHeader.
RQ1206_123	12.6.3.7.2	Upon reception of the "Si1.CancelPreparation" function command, the SPB Manager shall: • Return an error with the code eCodeMUnknown if a CodeM is provided as input data and is unknown to the SPB Manager.
RQ1206_124	12.6.3.7.2	Upon reception of the "Si1.CancelPreparation" function command, the SPB Manager shall: • Return an error with the code eCodeMNotAllowed if the bound SPB image download procedure as defined in clause 12.3.3.2 of ETSI TS 103 666-2 [10] associated with the Secondary Platform Bundle identifier linked to the CodeM provided as input data is completed.
RQ1206_125	12.6.3.7.2	Upon reception of the "Si1.CancelPreparation" function command, the SPB Manager shall: • Return an error with the code eSpbIdUnknown if a aSpbId is provided as input data and is unknown to the SPB Manager.
RQ1206_126	12.6.3.7.2	Upon reception of the "Si1.CancelPreparation" function command, the SPB Manager shall: • Return an error with the code eSpbIdNotAllowed if a aSpbId is provided as input data and is not linked with the CodeM provided as input data.
RQ1206_127	12.6.3.7.2	Upon reception of the "Si1.CancelPreparation" function command, the SPB Manager shall: • Cancel any pending procedure associated with the CodeM and/or the SpbId provided as input parameter(s), e.g. download procedure.
RQ1206_128	12.6.3.7.2	Upon reception of the "Si1.CancelPreparation" function command, the SPB Manager shall: • Unreserved the Secondary Platform Bundle identifier provided as input data and/or linked to the CodeM provided as input data.
RQ1206_129	12.6.3.7.2	Upon reception of the "Si1.CancelPreparation" function command, the SPB Manager shall: • Remove any reference to the CodeM if provided as input data.
RQ1206_130	12.6.3.7.2	Upon reception of the "Si1.CancelPreparation" function command, the SPB Manager shall: • Build Si1CancelPreparationResponse containing either an error code if one of the above step has failed or either the CodeM (aCodeM) or the Secondary Platform Bundle identifier (aSpbId) if provided as input data and the linked Secondary Platform identifier if any. Si1CancelPreparationResponse may also contain family identifier and/or service provider specific information. Their content is not in the scope of the present document.
RQ1206_131	12.6.3.7.2	Upon reception of the "Si1.CancelPreparation" function command, the SPB Manager shall: • Send the response to the service provider.
RQ1206_132	12.6.3.7.3	The body part of the HTTP POST response for the "Si1.CancelPreparation" function shall contain Si1CancelPreparationResponse defined as follows: • aSi1ResponseHeader: header of the response as defined in clause 12.6.3.2.2 of ETSI TS 103 666-2 [10]. • aCodeM: CodeM as it was in the incoming command.

Req.ID	Clause	Description
		• aSpbId: identifier of the Secondary Platform Bundle linked to aCodeM if aCodeM was provided as input data or aSpbId as if was in the incoming command. • aCustodianSpecificInfoBlock: specific parameter which may be defined by the custodian of the family identifier issuing the response. How this parameter is handled by the SPB Manager is out of scope of the present document. • aServiceProviderSpecificInfoBlock: specific parameter which may be defined by the service provider. How this parameter is handled by the SPB Manager is out of scope of the present document.
	12.6.3.8	***Si1.HandleNotification***
RQ1206_133	12.6.3.8.1	The "Si1.HandleNotification" function shall be used by the SPB Manager to send any notifications as agreed with the service provider owning the pending related task. The agreement of the notifications to send is outside the scope of the present document.
RQ1206_134	12.6.3.8.1	The body part of the HTTP POST request for the "Si1.HandleNotification" function shall contain Si1HandleNotificationBlock defined as follows: • aNotificationReceiverId: identifier of the recipient of the notification. It may equal to the function requester identity extracted from the last request-response function related to the same pending task, e.g. to the same download procedure. • aNotificationCallId: identifier of the function caller in the context of the recipient of the notification. It may be equal to the function caller identity extracted from the last request-response function related to the same pending task, e.g. to the same download procedure. • aCodeM: task's reference to cancel. This parameter shall be present if aSpbId is not provided as input parameters. • aSpbId: identifier of the Secondary Platform Bundle associated to the procedure to cancel. This parameter shall be present if aCodeM is not provided as input parameters. • aSpbType: type of Secondary Platform Bundle in which the SPB Manager shall select an available Secondary Platform Bundle identifier. • aPpIdentifier: identifier of the primary platform to link with the Secondary Platform Bundle reserved by the Si1.SelectSpb function. • aTimeStamp: indicates the date/time when the operation has been performed or when the notification has been received by the SPB Manager. • aNotificationEvent: indicates the step reached by the procedure that was executed. • aNotificationEventStatus: indicates the status after the execution of the notification. • aCustodianSpecificInfoBlock: specific parameter which may be defined by the custodian of the family identifier issuing the command. How this parameter is handled by the SPB Manager is out of scope of the present document. • aServiceProviderSpecificInfoBlock: specific parameter which may be defined by the service provider. How this parameter is handled by the SPB Manager is out of scope of the present document.
RQ1206_135	12.6.3.8.2	Table 12.6 ETSI TS 103 666-2 [10] indicates which parameters shall be present depending on aNotificationEvent.

NOTE: RQ1206_136 to RQ1206_139 are set to void due to numbering and duplication issues.

5.10.8 Interfaces and functions - Si2 interface

Reference: ETSI TS 103 666-2 [10] clause 12.6.4.

Req.ID	Clause	Description
	12.6.4.1	***Overview***
RQ1206_140	12.6.4.1	The Si2 interface is used between the LBA and SPB Manager to provide a transport of the bound Secondary Platform Bundle image and the management commands on the Secondary Platform Bundles installed in the iSSP.
RQ1206_141	12.6.4.1	The binding of the Si2 interface shall be performed over Hypertext Transfer Protocol version 2 (HTTP/2) as defined in IETF RFC 7540 [26] and the Transport Layer Security (TLS) version 1.3 in server authentication mode as defined in IETF RFC 8446 [27].
RQ1206_142	12.6.4.1	The LBA shall be in charge of managing the connection establishment to the SPB Manager for the Si2 interface.
RQ1206_143	12.6.4.1	The LBA shall use HTTP POST request message with HTTP path 'etsi/issp/si2/asn1' to deliver any function command over the Si2 interface.
	12.6.4.2	***Si2.GetSpbmCertificate***
RQ1206_144	12.6.4.2.1	The "Si2.GetSpbmCertificate" function shall be used by the LBA during the capability negotiation procedure as defined in clause 12.3.3.1 of ETSI TS 103 666-2 [10].
RQ1206_145	12.6.4.2.1	The LBA shall use the "Si2.GetSpbmCertificate" function to provide the SPB Manager with the public SSP information (SspInfoPublic) as defined in clause 12.6.2.2.2 of ETSI TS 103 666-2 [10] and terminal information (TerminalInfo) as defined in clause 12.6.2.7 of ETSI TS 103 666-2 [10].
RQ1206_146	12.6.4.2.1	The body part of the HTTP POST request for the "Si2.GetSpbmCertificate" function command shall contain Si2GetSpbmCertificateCommand with aSspInfoPublic - Public SSP information as defined in clause 12.6.2.2.2 of ETSI TS 103 666-2 [10] and aTerminalInfo - Terminal information as defined in clause 12.6.2.7 of ETSI TS 103 666-2 [10].
RQ1206_147	12.6.4.2.2	On reception of "Si2.GetSpbmCertificate" function command, the SPB Manager shall: 1) Perform eligibility check based on Annex C as follows: a) The SPB Manager shall verify that the aSpblSpecVerInfo contained in the aSspInfoPublic and aLbaSpecVerInfo contained in aTerminalInfo are supported by itself. If a version is not supported, the SPB Manager shall return eNotSupportedLbaVersion or eNotSupportedSpblVersion (the error indicating that the version of the Secondary Platform Bundle Loader or the LBA is not supported).
RQ1206_148	12.6.4.2.2	On reception of "Si2.GetSpbmCertificate" function command, the SPB Manager shall: 2) Determine the family identifier of the Secondary Platform Bundle container to be provisioned as follows: a) If the SPB Manager supports only one family identifier, the SPB Manager shall select that family identifier. If there is an aSspFamilyCryptoInfoBlock and no aSspGeneralCryptoInfo inside the aSspInfoPublic, the SPB Manager shall check whether one of the family identifiers contained in the aSspFamilyCryptoInfoBlock is supported. If supported, the SPB Manager shall select that family identifier. If not supported, the SPB Manager shall return eNotSupportedFamilyId (the error indicating that the family identifier is not supported). b) If the SPB Manager supports multiple family identifiers: – If there is only one SspFamilyCryptoInfoBlock data structure containing a family identifier supported by the SPB Manager, the SPB Manager shall select that family identifier. If there is no SspFamilyCryptoInfoBlock data structure containing a family identifier supported by the SPB Manager, the SPB Manager shall return eNotSupportedFamiyId (the error indicating that the family identifier is not supported). – If there are multiple aSspFamilyCryptoInfoBlock data structure containing the family identifier supported by the SPB Manager inside the aSspInfoPublic or there is only aSspCryptoInfo inside the aSspInfoPublic, the SPB Manager shall return eSpblSelectOneFamilyId (the error indicating that one family identifier shall be selected by the Secondary Platform Bundle Loader).
RQ1206_148a	12.6.4.2.2	On reception of "Si2.GetSpbmCertificate" function command, the SPB Manager shall: 2) Determine the family identifier of the Secondary Platform Bundle container to be provisioned as follows: a) If the SPB Manager supports only one family identifier, the SPB Manager shall select that family identifier. If there is an aSspFamilyCryptoInfoBlock and no aSspGeneralCryptoInfo inside the aSspInfoPublic, the SPB Manager shall check whether one of the family identifiers contained in the aSspFamilyCryptoInfoBlock is supported. If supported, the SPB Manager shall select that family identifier. If not supported, the SPB Manager shall return eNotSupportedFamilyId (the error indicating that the family identifier is not supported).

Req.ID	Clause	Description
RQ1206_148b	12.6.4.2.2	On reception of "Si2.GetSpbmCertificate" function command, the SPB Manager shall: 2) Determine the family identifier of the Secondary Platform Bundle container to be provisioned as follows: b) If the SPB Manager supports multiple family identifiers: – If there is only one SspFamilyCryptoInfoBlock data structure containing a family identifier supported by the SPB Manager, the SPB Manager shall select that family identifier. If there is no SspFamilyCryptoInfoBlock data structure containing a family identifier supported by the SPB Manager, the SPB Manager shall return eNotSupportedFamiyId (the error indicating that the family identifier is not supported).
RQ1206_148c	12.6.4.2.2	On reception of "Si2.GetSpbmCertificate" function command, the SPB Manager shall: 2) Determine the family identifier of the Secondary Platform Bundle container to be provisioned as follows b) If the SPB Manager supports multiple family identifiers: – If there are multiple aSspFamilyCryptoInfoBlock data structure containing the family identifier supported by the SPB Manager inside the aSspInfoPublic or there is only aSspCryptoInfo inside the aSspInfoPublic, the SPB Manager shall return eSpblSelectOneFamilyId (the error indicating that one family identifier shall be selected by the Secondary Platform Bundle Loader).
RQ1206_149	12.6.2.2.2	On reception of "Si2.GetSpbmCertificate" function command, the SPB Manager shall: 3) Set the selected family identifier into the aSpbFamilyId.
RQ1206_150	12.6.2.2.2	On reception of "Si2.GetSpbmCertificate" function command, the SPB Manager shall: 4) Using the selected family identifier, select one of aSspCryptoInfo, aSspFamilyCrytoInfo and aSspOidCryptoInfo inside the aSspInfoPublic as follows: a) If there is a SspFamilyCryptoInfoBlock data structure containing the selected family identifier, the SPB Manager shall select that SspFamilyCryptoInfoBlock data structure. Using the selected SspFamilyCryptoInfoBlock data structure: – If the SPB Manager supports only one custodian for the selected family identifier and there is a SspOidCryptoInfoBlock data structure containing the OID of that custodian, the SPB Manager shall select the aSspOidCyrptoInfo data structure contained in that SspOidCryptoInfoBlock data structure. If there is no SspOidCryptoInfoBlock data structure containing the OID of that custodian, the SPB Manager shall select the aSspFamilyCryptoInfo inside the SspFamilyCryptoInfoBlock data structure. – If the SPB Manager supports multiple custodians for the selected family identifier and there is only one SspOidCryptoInfoBlock data structure containing the Oid of one of the custodians supported by the SPB Manager, the SPB Manager shall select the aSspOidCryptoInfo contained in that SspOidCryptoInfoBlock data structure. – If the SPB Manager supports multiple custodians for the selected family identifier and there are multiple SspOidCryptoInfoBlock data structures containing the OIDs of custodians supported by the SPB Manager, the SPB Manager shall return eSpblSelectOneOid (the error indicating that one custodian shall be selected by the Secondary Platform Bundle Loader). – If there is no SspOidCryptoInfo data structure containing the OID of a custodian supported by the SPB Manager, the SPB Manager shall select the aSspFamilyCryptoInfo. b) If there is no SspFamilyCryptoInfoBlock data structure containing the selected family identifier, the SPB Manager shall select the aSspGeneralCryptoInfo inside the aSspInfoPublic.
RQ1206_151	12.6.4.2.2	On reception of "Si2.GetSpbmCertificate" function command, the SPB Manager shall: 5) Using the selected SspCryptoInfo data structure, choose the following: a) An SPB Manager certificate for key agreement that can be verified by the trusted public key indicated by one of the trusted public key identifiers in the aSspPkIdListForSpbmVerification. If none of the trusted public key identifiers in the aSspPkIdListForSpbmVerification is supported, the SPB Manager shall return eNotSupportedPkIdSpbmVerification. The algorithmIdentifier of the selected certificate shall be one of the algorithmIdentifier in aKeyAgreementAlgorithmList. If the algorithmIdentifier of the selected certificate is not supported, the SPB Manager shall return eNotSupportedKeyAgreementAlgorithm.

Req.ID	Clause	Description
		b) An SPB Manager certificate for digital signature that can be verified by the trusted public key indicated by one of the trusted public key identifiers in the aSspPkIdListForSpbmVerification. If any trusted public key identifiers in the aSspPkIdListForSpbmVerification is not supported, the SPB Manager shall return eNotSupportedPkIdSpbmVerification. c) One of trusted public key identifier(s) in the aSspPkIdListForSpblVerification that shall be used by the Secondary Platform Bundle Loader to select its certificate(s). The SPB Manager shall set the selected trusted public key identifier into aSspCiPkIdToBeUsed. If any trusted public key identifiers in the aSspPkIdListForSpblVerification is not supported, the SPB Manager shall return eNotSupportedPkIdSpblVerification. d) d) One of algorithmIdentifiers in the aCipherAlgorithmList that shall be used by the Secondary Platform Bundle Loader and the SPB Manager for data encryption. The SPB Manager shall set the selected algorithmIdentifier into aSspCryptoToBeUsed. If none of the algorithmIdentifier in the aCipherAlgorithmList is supported, the SPB Manager shall return eNotSupportedEncryptionAlgorithm.
RQ1206_151a	12.6.4.2.2	On reception of "Si2.GetSpbmCertificate" function command, the SPB Manager shall: 5) Using the selected SspCryptoInfo data structure, choose the following: a) An SPB Manager certificate for key agreement that can be verified by the trusted public key indicated by one of the trusted public key identifiers in the aSspPkIdListForSpbmVerification. If none of the trusted public key identifiers in the aSspPkIdListForSpbmVerification is supported, the SPB Manager shall return eNotSupportedPkIdSpbmVerification.The algorithmIdentifier of the selected certificate shall be one of the algorithmIdentifier in aKeyAgreementAlgorithmList. If the algorithmIdentifier of the selected certificate is not supported, the SPB Manager shall return eNotSupportedKeyAgreementAlgorithm.
RQ1206_151b	12.6.4.2.2	On reception of "Si2.GetSpbmCertificate" function command, the SPB Manager shall: 5) Using the selected SspCryptoInfo data structure, choose the following: b) An SPB Manager certificate for digital signature that can be verified by the trusted public key indicated by one of the trusted public key identifiers in the aSspPkIdListForSpbmVerification. If any trusted public key identifiers in the aSspPkIdListForSpbmVerification is not supported, the SPB Manager shall return eNotSupportedPkIdSpbmVerification.
RQ1206_151c	12.6.4.2.2	On reception of "Si2.GetSpbmCertificate" function command, the SPB Manager shall: 5) Using the selected SspCryptoInfo data structure, choose the following: c) One of trusted public key identifier(s) in the aSspPkIdListForSpblVerification that shall be used by the Secondary Platform Bundle Loader to select its certificate(s). The SPB Manager shall set the selected trusted public key identifier into aSspCiPkIdToBeUsed. If any trusted public key identifiers in the aSspPkIdListForSpblVerification is not supported, the SPB Manager shall return eNotSupportedPkIdSpblVerification.
RQ1206_151d	12.6.4.2.2	On reception of "Si2.GetSpbmCertificate" function command, the SPB Manager shall: 5) Using the selected SspCryptoInfo data structure, choose the following: d) One of algorithmIdentifiers in the aCipherAlgorithmList that shall be used by the Secondary Platform Bundle Loader and the SPB Manager for data encryption. The SPB Manager shall set the selected algorithmIdentifier into aSspCryptoToBeUsed. If none of the algorithmIdentifier in the aCipherAlgorithmList is supported, the SPB Manager shall return eNotSupportedEncryptionAlgorithm.
RQ1206_152	12.6.4.2.2	On reception of "Si2.GetSpbmCertificate" function command, the SPB Manager shall: 6) Generate a new random octet string for aChallengeS which shall be used to authenticate the Secondary Platform Bundle Loader.
RQ1206_153	12.6.4.2.2	On reception of "Si2.GetSpbmCertificate" function command, the SPB Manager shall: 7) Build Si2GetSpbmCertificateResponse containing the SPB Manager certificate for key agreement, the aSspCiPkIdToBeUsed, the aSspCryptoToBeUsed, aChallengeS, and the aSpbFamilyId and optionally the certificate chain for SPB Manager certificate for key agreement.
RQ1206_154	12.6.4.2.3	The body part of the HTTP POST response for the "Si2.GetSpbmCertificate" function shall contain Si2GetSpbmCertificateResponse defined as follows: • aSspPkIdForSpblVerification: CI Public Key identifier for SPBL Certificate which shall be used by the Secondary Platform Bundle Loader for signature generation. • aSspCryptoToBeUsed: algorithm identifiers for data encryption which shall be used by the Secondary Platform Bundle Loader and the SPB Manager. • aSpbmKaCert: SPBM certificate for key agreement.

Req.ID	Clause	Description
		• aSpbFamilyId: the family identifier of the Secondary Platform Bundle. • CustodianOid: the OID of a custodian for the aSpbFamilyId. • aChallengeS: the value of CHALLENGE_S as defined in GlobalPlatform Open Firmware Loader for Tamper Resistant Element [13]. The aChallengeS is generated by the SPB Manager and used in authentication of the Secondary Platform Bundle Loader. • aSpbmCertChain: the certificates to be used to construct certification path for verification of SPBM certificate for key agreement.
	12.6.4.3	***Si2.GetBoundSpbImage***
RQ1206_155	12.6.4.3.1	The "Si2.GetBoundSpbImage" function shall be used by the LBA during the download procedure as defined in clause 12.3.3 of ETSI TS 103 666-2 [10].
RQ1206_156	12.6.4.3.1	The LBA shall use "Si2.GetBoundSpbImage" function to provide the SPB Manager with SSP credential (SspCredential) as defined in clause 12.6.2.4 of ETSI TS 103 666-2 [10] and terminal information (TerminalInfo) as defined in clause 12.6.2.7 of ETSI TS 103 666-2 [10].
RQ1206_157	12.6.4.3.1	The LBA shall provide RequestType to the SPB Manager to indicate the request type. The LBA shall set: • "RequestBoundSpbImage" to the RequestType if the LBA requests a bound Secondary Platform Bundle image. • "RequestSpbMetadata" to the RequestType if the LBA requests only SPB metadata before requesting a bound Secondary Platform Bundle image to check the SPB metadata and, if configured, require the user intent. • "BoundSpbImageByTransacId" to the RequestType if the LBA requests a bound Secondary Platform Bundle image after receiving the SPB metadata via "Si2.GetBoundSpbImage" function with "RequestSpbMetadata" as the requestType.
RQ1206_158	12.6.4.3.1	The body part of the HTTP POST request for the "Si2.GetBoundSpbImage" function command shall contain Si2GetBoundSpbImageCommand defined as follows: • aSspCredential: – SSP credential as defined in clause 12.6.2.4 of ETSI TS 103 666-2 [10] • aTerminalInfo: – Terminal Information as defined in clause 12.6.2.7 of ETSI TS 103 666-2 [10] • aRequestType: – eRequestBoundSpbImage – eRequestSpbMetadata – eBoundSpbImageByTransacId
RQ1206_159	12.6.4.3.2	On reception of the Si2.GetBoundSpbImage function command, the SPB Manager shall check the value of the aRequestType contained in the Si2GetBoundSpbImage and perform the procedure as described below: 1) If the value of aRequestType is "eRequestBoundSpbImage (0)" or "eRequestSpbMetadata (1)", the SPB Manager shall: a) Extract the aSpbmKaPkIdToBeUsed contained in the aSspImageSessionToken in the aSspCredential.
RQ1206_160	12.6.4.3.2	On reception of the Si2.GetBoundSpbImage function command, the SPB Manager shall check the value of the aRequestType contained in the Si2GetBoundSpbImage and perform the procedure as described below: 1) If the value of aRequestType is "eRequestBoundSpbImage (0)" or "eRequestSpbMetadata (1)", the SPB Manager shall: b) Selecta aSpbmKaCertificate which can be verified by the CI certificate indicated in aSpbmKaPkIdToBeUsed.
RQ1206_161	12.6.4.3.2	On reception of the Si2.GetBoundSpbImage function command, the SPB Manager shall check the value of the aRequestType contained in the Si2GetBoundSpbImage and perform the procedure as described below: 1) If the value of aRequestType is "eRequestBoundSpbImage (0)" or "eRequestSpbMetadata (1)", the SPB Manager shall: c) Generate the first session key as defined in GlobalPlatform Open Firmware Loader for Tamper Resistant Element [13]. The first session key shall be generated by using the private key corresponding to the SPB Manager certificate for key agreement and the aEPkSpbIKa contained in the aTbsSspImageSessionToken in the aSspCredential. NOTE: The first session key is the same as 'KS1' in GlobalPlatform Open Firmware Loader for Tamper Resistant Element [13].

Req.ID	Clause	Description
RQ1206_162	12.6.4.3.2	On reception of the Si2.GetBoundSpbImage function command, the SPB Manager shall check the value of the aRequestType contained in the Si2GetBoundSpbImage and perform the procedure as described below: 1) If the value of aRequestType is "eRequestBoundSpbImage (0)" or "eRequestSpbMetadata (1)", the SPB Manager shall: d) Decrypt the aM-SSP contained in the aSspCredential by using the first session key. The SPB Manager shall use the algorithm identified by the aSspCryptoToBeUsed. The SPB Manager shall obtain the Secondary Platform Bundle Loader certificate, aTbsSspToken, and the signature of TbsSspToken by decrypting the aM-SSP.
RQ1206_163	12.6.4.3.2	On reception of the Si2.GetBoundSpbImage function command, the SPB Manager shall check the value of the aRequestType contained in the Si2GetBoundSpbImage and perform the procedure as described below: 1) If the value of aRequestType is "eRequestBoundSpbImage (0)" or "eRequestSpbMetadata (1)", the SPB Manager shall: e) Verify the Secondary Platform Bundle Loader certificate by using the trust public key which is identified by the aSspPkIdForSpblVerification. The Secondary Platform Bundle Loader certificate shall be verified based on the certification path verification as defined in clause 12.2.1.1.4 of ETSI TS 103 666-2 [10]. If the verification fails, the SPB Manager shall return eInvalidSpblCertificate.
RQ1206_164	12.6.4.3.2	On reception of the Si2.GetBoundSpbImage function command, the SPB Manager shall check the value of the aRequestType contained in the Si2GetBoundSpbImage and perform the procedure as described below: 1) If the value of aRequestType is "eRequestBoundSpbImage (0)" or "eRequestSpbMetadata (1)", the SPB Manager shall: f) Verify the aSspImageSessionTokenSignature and aTbsSspTokenSignature by using the Secondary Platform Bundle Loader certificate.
RQ1206_165	12.6.4.3.2	On reception of the Si2.GetBoundSpbImage function command, the SPB Manager shall check the value of the aRequestType contained in the Si2GetBoundSpbImage and perform the procedure as described below: 1) If the value of aRequestType is "eRequestBoundSpbImage (0)" or "eRequestSpbMetadata (1)", the SPB Manager shall: g) Store the aIdTransac contained in the aTbsSspImageSessionToken and attach aChallengeS to the aIdTransac to manage this on-going image session.
RQ1206_166	12.6.4.3.2	On reception of the Si2.GetBoundSpbImage function command, the SPB Manager shall check the value of the aRequestType contained in the Si2GetBoundSpbImage and perform the procedure as described below: 1) If the value of aRequestType is "eRequestBoundSpbImage (0)" or "eRequestSpbMetadata (1)", the SPB Manager shall: h) Find the Secondary Platform Bundle identifier corresponding to the aCodeM contained in the aTbsSspToken. If there is not the Secondary Platform Bundle identifier corresponding to aCodeM, the SPB Manager shall call the "Si1.HandleNotification" function. The function command shall contain the aNotificationEvent, the aCodeM, the aSspInfoProtected and aTerminalInfo. The aNotificationEvent shall be set to eNotificationStatus_Eligibility. The SPB Manager shall suspend the bound SPB image download procedure until the service provider has completed the Secondary Platform Bundle selection process as defined in clause 12.3.2.2 of ETSI TS 103 666-2 [10].
RQ1206_167	12.6.4.3.2	On reception of the Si2.GetBoundSpbImage function command, the SPB Manager shall check the value of the aRequestType contained in the Si2GetBoundSpbImage and perform the procedure as described below: 1) If the value of aRequestType is "eRequestBoundSpbImage (0)" or "eRequestSpbMetadata (1)", the SPB Manager shall: i) Perform the eligibility check based on Annex C by using the aSspInfoProtected contained in aTbsSspToken and aTermianlInfo. The SPB Manager shall: • Verify the aPpIdentifier contained in the aSspInfoProtected. • Verify the aFamilySpecificSspInfoBlock contained in the aSspInfoProtected (out of scope of the present document). • Verify the aFamilySpecificTerminalInfo and aOidSpecificInfo contained in the aTermianlInfo (out of scope of the present document). • Check whether the selected Secondary Platform Bundle image is supported by the iSSP based on aPpIdentifier, aSspInfoProtected, and aTerminalInfo. If the selected Secondary Platform Bundle image is not supported, the SPB Manager shall return eInvalidSpbImage.

Req.ID	Clause	Description
RQ1206_168	12.6.4.3.2	On reception of the Si2.GetBoundSpbImage function command, the SPB Manager shall check the value of the aRequestType contained in the Si2GetBoundSpbImage and perform the procedure as described below: 1) If the value of aRequestType is "eRequestBoundSpbImage (0)" or "eRequestSpbMetadata (1)", the SPB Manager shall: j) Build aSpbMetadata corresponding to the selected Secondary Platform Bundle image (out of scope of the present document). The SPB Manager may construct aFamilySpecificData and aOidSpecificData in aSpbMetadata based on aFamilySpecificSspInfoBlock and aFamilySpecificTerminalInfoBlock contained in Si2GetBoundSpbImageCommand.
RQ1206_169	12.6.4.3.2	On reception of the Si2.GetBoundSpbImage function command, the SPB Manager shall check the value of the aRequestType contained in the Si2GetBoundSpbImage and perform the procedure as described below: 1) If the value of aRequestType is "eRequestBoundSpbImage (0)" or "eRequestSpbMetadata (1)", the SPB Manager shall: k) If the aRequestType is "eRequestSpbMetadata (1)", the SPB Manager shall: • Bind the aSspCredential to the aIdTransac. • Return aSpbMetadata to the LBA as the response of the "Si2.GetBoundSpbImage" function.
RQ1206_170	12.6.4.3.2	On reception of the Si2.GetBoundSpbImage function command, the SPB Manager shall check the value of the aRequestType contained in the Si2GetBoundSpbImage and perform the procedure as described below: 2) If the aRequestType is " eBoundSpbImageByTransacId (2)", the SPB Manager shall verify that the aSspCredential is verified in step 1 with aRequestType set to "eRequestSpbMetadata (1)". If the verification fails, the SPB Manager shall return eInvalidBoundSpbImageByTransacId.
RQ1206_171	12.6.4.3.2	On reception of the Si2.GetBoundSpbImage function command, the SPB Manager shall check the value of the aRequestType contained in the Si2GetBoundSpbImage and perform the procedure as described below: 3) After successfully finishing the above steps, the SPB Manager shall: a) Generate TIME_STAMP and generate aM-TimeStamp by encrypting the TIME_STAMP by using the first session key and the encryption algorithm identified by the aSspCyrptoToBeUsed determined in the capability negotiation procedure as defined in clause 12.3.3.1 of ETSI TS 103 666-2 [10].
RQ1206_172	12.6.4.3.2	On reception of the Si2.GetBoundSpbImage function command, the SPB Manager shall check the value of the aRequestType contained in the Si2GetBoundSpbImage and perform the procedure as described below: 3) After successfully finishing the above steps, the SPB Manager shall: b) Generate an SPB Manager's ephemeral key pair. The domain parameter used to generate the ephemeral key pair shall be the same as the one used by the SPB Manager certificate for key agreement.
RQ1206_173	12.6.4.3.2	On reception of the Si2.GetBoundSpbImage function command, the SPB Manager shall check the value of the aRequestType contained in the Si2GetBoundSpbImage and perform the procedure as described below: 3) After successfully finishing the above steps, the SPB Manager shall: c) Generate the second session key as defined in GlobalPlatform Open Firmware Loader for Tamper Resistant Element [13]. The second session key shall be generated with the SPB Manager's ephemeral private key and the aEPkSpbIKa contained in aTbsSspImagesessionToken. NOTE: The second session key is the same as 'KS2' in GlobalPlatform Open Firmware Loader for Tamper Resistant Element [13].
RQ1206_174	12.6.4.3.2	On reception of the Si2.GetBoundSpbImage function command, the SPB Manager shall check the value of the aRequestType contained in the Si2GetBoundSpbImage and perform the procedure as described below: 3) After successfully finishing the above steps, the SPB Manager shall: d) Generate aSpbmToken data structure containing the SPB Manager's ephemeral public key and the aIdTransac as defined in clause 12.6.2.5 of ETSI TS 103 666-2 [10].

Req.ID	Clause	Description
RQ1206_175	12.6.4.3.2	On reception of the Si2.GetBoundSpbImage function command, the SPB Manager shall check the value of the aRequestType contained in the Si2GetBoundSpbImage and perform the procedure as described below: 3) After successfully finishing the above steps, the SPB Manager shall: e) Select an SPB Manager certificate for digital signature which can be verified by the trusted public key indicated by one of the trusted public key identifiers in the aSspPkIdListForSpbmVerification. The selected SPB Manager certificate for digital signature shall be verified by the same trusted public key as the one used to verify the SPB Manager certificate for key agreement determined by the "Si2.GetSpbmCertificate" function.
RQ1206_176	12.6.4.3.2	On reception of the Si2.GetBoundSpbImage function command, the SPB Manager shall check the value of the aRequestType contained in the Si2GetBoundSpbImage and perform the procedure as described below: 3) After successfully finishing the above steps, the SPB Manager shall: f) Compute the aSpbmTokenSignature over the aTbsSpbmToken using the private key corresponding to the SPB Manager certificate for digital signature.
RQ1206_177	12.6.4.3.2	On reception of the Si2.GetBoundSpbImage function command, the SPB Manager shall check the value of the aRequestType contained in the Si2GetBoundSpbImage and perform the procedure as described below: 3) After successfully finishing the above steps, the SPB Manager shall: g) Obtain aM-IMD by encrypting the Image Descriptor (IMD) by using the second session key as defined in GlobalPlatform Open Firmware Loader for Tamper Resistant Element [13].
RQ1206_178	12.6.4.3.2	On reception of the Si2.GetBoundSpbImage function command, the SPB Manager shall check the value of the aRequestType contained in the Si2GetBoundSpbImage and perform the procedure as described below: 3) After successfully finishing the above steps, the SPB Manager shall: h) Obtain aM-ARP by encrypting the ATK.ARP.ECDSA by using the second session key as defined in GlobalPlatform Open Firmware Loader for Tamper Resistant Element [13].
RQ1206_179	12.6.4.3.2	On reception of the Si2.GetBoundSpbImage function command, the SPB Manager shall check the value of the aRequestType contained in the Si2GetBoundSpbImage and perform the procedure as described below: 3) After successfully finishing the above steps, the SPB Manager shall: i) Build aDoOperateParameter data structure as defined in clause 12.6.2.5 of ETSI TS 103 666-2 [10].
RQ1206_180	12.6.4.3.2	On reception of the Si2.GetBoundSpbImage function command, the SPB Manager shall check the value of the aRequestType contained in the Si2GetBoundSpbImage and perform the procedure as described below: 3) After successfully finishing the above steps, the SPB Manager shall: j) Build aChangeSegmentParameters data structure as defined in clause 12.6.2.5 of ETSI TS 103 666-2 [10]. The aChangeSegmentParameters shall be the list of ChangeSegmentParameters. Each ChangeSegmentParameter shall be generated by encrypting the Segment Descriptor Structure by using the second session key as defined in GlobalPlatform Open Firmware Loader for Tamper Resistant Element [13].
RQ1206_181	12.6.4.3.2	Void
RQ1206_182	12.6.4.3.2	On reception of the Si2.GetBoundSpbImage function command, the SPB Manager shall check the value of the aRequestType contained in the Si2GetBoundSpbImage and perform the procedure as described below: 3) After successfully finishing the above steps, the SPB Manager shall: k) Build aBoundSpbImage data structure as defined in clause 12.6.2.5. of ETSI TS 103 666-2 [10].
RQ1206_183	12.6.4.3.2	On reception of the Si2.GetBoundSpbImage function command, the SPB Manager shall check the value of the aRequestType contained in the Si2GetBoundSpbImage and perform the procedure as described below: 3) After successfully finishing the above steps, the SPB Manager shall: l) Return the aBoundSpbImage data structure to the LBA as the response of the "Si2.GetBoundSpbImage" function.
RQ1206_184	12.6.4.3.3	The body part of the HTTP POST response of the "Si2.GetBoundSpbImage" shall contain Si2GetBoundSpbImageResponse defined as follows: • aBoundSpbImage: Secondary Platform Bundle image bound to the Secondary Platform Bundle Loader. • aSpbMetadata: the SPB metadata of the Secondary Platform Bundle corresponding to the aCodeM. • aSi2GetBoundSpbImageErrorCode:

Req.ID	Clause	Description
		– eInvalidSpblCertificate: the error indicating that the SPBL certification path could not be verified. – eInvalidCodeM: the error indicating that the aCodeM has not been reserved by the Service Provider. – eInvalidChallengeS: the error indicating that aChallengeS is not valid in this image session. – eInvalidSpbImage: the error indicating that the Secondary Platform Bundle corresponding to the aCodeM is not compatible with the SSP. – eInvalidBoundSpbImageByTransacId: the error indicating that the aSspCredential containing this aIdTransac has not been verified with aRequestType set to "eRequestSpbMetadata (1)". – aSpbFamilyId: the family identifier of the Secondary Platform Bundle referenced by the aCodeM. – aFamilySpecificError: a family identifier-specific error container which may be defined for the aSpbFamilyId. – aOidSpecificError: a family identifier-specific error container which may be defined by the custodian indicated in aOid contained in aOidSpecificError.
RQ1206_184a	12.6.4.3.3	The body part of the HTTP POST response of the "Si2.GetBoundSpbImage" may contain error codes in the Si2GetBoundSpbImageResponse as defined as follows: • aSi2GetBoundSpbImageErrorCode: – eInvalidSpblCertificate: the error indicating that the SPBL certification path could not be verified.
RQ1206_184b	12.6.4.3.3	The body part of the HTTP POST response of the "Si2.GetBoundSpbImage" may contain error codes in the Si2GetBoundSpbImageResponse as defined as follows: • aSi2GetBoundSpbImageErrorCode: – eInvalidCodeM: the error indicating that the aCodeM has not been reserved by the Service Provider.
RQ1206_184c	12.6.4.3.3	The body part of the HTTP POST response of the "Si2.GetBoundSpbImage" may contain error codes in the Si2GetBoundSpbImageResponse as defined as follows: • aSi2GetBoundSpbImageErrorCode: – eInvalidChallengeS: the error indicating that aChallengeS is not valid in this image session.
RQ1206_184d	12.6.4.3.3	The body part of the HTTP POST response of the "Si2.GetBoundSpbImage" may contain error codes in the Si2GetBoundSpbImageResponse as defined as follows: • aSi2GetBoundSpbImageErrorCode: – eInvalidSpbImage: the error indicating that the Secondary Platform Bundle corresponding to the aCodeM is not compatible with the SSP.
RQ1206_184e	12.6.4.3.3	The body part of the HTTP POST response of the "Si2.GetBoundSpbImage" may contain error codes in the Si2GetBoundSpbImageResponse as defined as follows: • aSi2GetBoundSpbImageErrorCode: – eInvalidBoundSpbImageByTransacId: the error indicating that the aSspCredential containing this aIdTransac has not been verified with.
RQ1206_184f	12.6.4.3.3	The body part of the HTTP POST response of the "Si2.GetBoundSpbImage" may contain error codes in the Si2GetBoundSpbImageResponse as defined as follows: • aFamilySpecificError: a family identifier-specific error container which may be defined for the aSpbFamilyId.
RQ1206_184g	12.6.4.3.3	The body part of the HTTP POST response of the "Si2.GetBoundSpbImage" may contain error codes in the Si2GetBoundSpbImageResponse as defined as follows: • aOidSpecificError: a family identifier-specific error container which may be defined by the custodian indicated in aOid contained in aOidSpecificError.
	12.6.4.4	***Si2.HandleNotification***
RQ1206_185	12.6.4.4.1	The "Si2.HandleNotification" function shall be used by the LBA to send any notification about the result of the Secondary Platform Bundle management to the SPB Manager.

Req.ID	Clause	Description
RQ1206_186	12.6.4.4.1	The body part of the HTTP POST request for the "Si2.HandleNotification" function command shall contain Si2HandleNotificationCommand defined as follows: • aNotificationEvent: it indicates the procedure related to this notification. • aTimeStamp: it indicates the time when this notification message is constructed by the LBA. • aSpbId: identifier of the Secondary Platform Bundle related to aNotificationEvent. • aNotificationToken: notification token which contains the information about the state change of the Secondary Platform Bundle container in the iSSP as defined in clause 12.6.2.8 of ETSI TS 103 666-2 [10]. • aCodeM: the CodeMatching identifier linked with the Secondary Platform Bundle to download. If the Si2NotificationEvent is 'eNotificationStatus_SPBInstallationError', this parameter shall be present. • aFamilySpecificNotificationCommand: family identifier-specific Si2HandleNotificationCommand which may be defined for that family identifier. How this parameter is handled by the SPB Manager is out of scope of the present document. • aCustodianSpecificNotificationCommand: Custodian-specific Si2HandleNotificationCommand which may be defined by a custodian identified by aOid inside the aCustodianSpecificNotificationCommand.
RQ1206_187	12.6.4.4.2	On reception of Si2HandleNotificationCommand, the SPB Manager shall respond to the LBA to notify a successful reception of the notification. The response may contain a family identifier-specific notification response or a custodianspecific notification response.
RQ1206_188	12.6.4.4.3	The body part of the HTTP POST response for the "Si2.HandleNotification" function shall contain Si2HandleNotificationResponse defined as follows: • aFamilySpecificNotificationResponse: a family identifier-specific Si2HandleNotificationResponse which may be defined for that family identifier. • aCustodianSpecificNotificationResponse: a custodian-specific Si2HandleNotificationResponse which may be defined by the custodian identified by aOid inside the aCustodianSpecificNotificationResponse.

5.10.9 Interfaces and functions - Si3 interface

Reference: ETSI TS 103 666-2 [10] clause 12.6.5.

Req.ID	Clause	Description
	12.6.5.1	***Overview***
RQ1206_274	12.6.5.1	The Si3 interface is used between the LBA and the Secondary Platform Bundle Loader. The LBA shall use the Si3 interface to transfer a bound Secondary Platform Bundle image and management commands to the Secondary Platform Bundle Loader.
RQ1206_189	12.6.5.1	The OFL agent host in the LBA and the OFL service hosted in the Secondary Platform Bundle Loader shall exchange commands, responses, and events over the Si3 interface as defined in as defined in GlobalPlatform VPP - OFL VNP Extension [16] with the additional commands, responses and registry defined in clauses 12.6.5.2, 12.6.5.3 and 12.6.5.4 of ETSI TS 103 666-2 [10].
RQ1206_190	12.6.5.2	The OFL service Gate in the Secondary Platform Bundle Loader shall support the commands defined in clause 7.3.1.3 of ETSI TS 103 666-2 [10].
RQ1206_191	12.6.5.3	The OFL service Gate in the Secondary Platform Bundle Loader shall support the commands defined in clause 7.3.1.3 of ETSI TS 103 666-2 [10].
RQ1206_192	12.6.5.4	The OFL service Gate in the Secondary Platform Bundle Loader shall support the responses defined in clause 7.3.1.4 of ETSI TS 103 666-2 [10].
	12.6.5.5	***Functions***
	12.6.5.5.1	***Si3.GetSspInfo***
RQ1206_193	12.6.5.5.1	The "Si3.GetSspInfo" function shall be used by the LBA during the capability negotiation procedure as defined in clause 12.3.3.1 of ETSI TS 103 666-2 [10].
RQ1206_194	12.6.5.5.1	The LBA shall use the "Si3.GetSspInfo" function to retrieve aSspInfoPublic from the Secondary Platform Bundle Loader.

Req.ID	Clause	Description
RQ1206_195	12.6.5.5.1	The "Si3.GetSspInfo" function command shall be GET_SSP_INFO. The parameter of GET_SSP_INFO command is defined as follows: On reception of the "Si3.GetSspInfo" function command, the Secondary Platform Bundle Loader shall: 1) Set the GET_SSP_INFO command parameter into the GET_SSP_INFO_PARAMETER registry.
RQ1206_196	12.6.5.5.1	The parameter of GET_SSP_INFO command is defined as follows: On reception of the "Si3.GetSspInfo" function command, the Secondary Platform Bundle Loader shall: 2) Build aSspInfoPublic data structure defined in clause 12.6.2.2.2 of ETSI TS 103 666-2 [10] as follows: a) The Secondary Platform Bundle Loader shall set the release of the specification that is implemented by the Secondary Platform Bundle Loader into the aSpblSpecVerInfo.
RQ1206_197	12.6.5.5.1	The parameter of GET_SSP_INFO command is defined as follows: On reception of the "Si3.GetSspInfo" function command, the Secondary Platform Bundle Loader shall: 2) Build aSspInfoPublic data structure defined in clause 12.6.2.2.2 of ETSI TS 103 666-2 [10] as follows: b) If GET_SSP_INFO contains both the aSpbFamilyId and the aCustodianOid, the Secondary Platform Bundle Loader shall build aSspInfoPublic containing: • One aSspFamilyCryptoInfoBlock which shall contain the aSpbFamilyId and only one aSspOidCryptoInfoBlock if there is a configuration for both of the aSpbFamilyId and the aOid. The aSspOidCryptoInfoBlock shall have aCustodianOid and aSspOidCryptoInfo which contains the list of trusted public key identifiers and the list of algorithm identifiers which are allowed to be used for loading of the Secondary Platform Bundles with that aSpbFamilyId and that aCustodianOid. • One aSspFamilyCryptoInfoBlock which shall contain the aSpbFamilyId and aSspFamilyCryptoInfo if there is a configuration for the aSpbFamilyId but not for the aCustodianOid. The aSspFamilyCryptoInfo shall contain the list of trusted public key identifiers and the list of algorithm identifiers which are allowed to be used for loading of the Secondary Platform Bundles with that aSpbFamilyId. • aSspGeneralCryptoInfo if there is no configuration for the aSpbFamilyId. The aSspGeneralCryptoInfo shall contain the list of trusted public key identifiers and the list of algorithm identifiers which are not associated with any family identifier and any custodian.
RQ1206_198	12.6.5.5.1	The parameter of GET_SSP_INFO command is defined as follows: On reception of the "Si3.GetSspInfo" function command, the Secondary Platform Bundle Loader shall: 2) Build aSspInfoPublic data structure defined in clause 12.6.2.2.2 of ETSI TS 103 666-2 [10] as follows: c) If GET_SSP_INFO command parameter contains only aSpbFamilyId, the Secondary Platform Bundle Loader shall build aSspInfoPublic containing: • One aSspFamilyCryptoInfoBlock which shall contain the aSpbFamilyId if there is a configuration for the aSpbFamilyId. The aSspFamilyCryptoInfoBlock may contain the set of aSspOidCryptoInfoBlocks as many as the configurations for the custodians of that aSpbFamilyId. Each aSspOidCryptoInfoBlock shall have aCustodianOid and aSspOidCryptoInfo which contains the list of trusted public key identifiers and the list of algorithm identifiers which are allowed to be used for loading of the Secondary Platform Bundles with that aSpbFamilyId and that aCustodianOid. The aSspFamilyCryptoInfoBlock may also contain aSspFamilyCryptoInfo. • aSspGeneralCryptoInfo if there is no configuration for the aSpbFamilyId. The aSspGeneralCryptoInfo shall contain the list of trusted public key identifiers and the list of algorithm identifiers which are not associated with any family identifier and any custodian.

Req.ID	Clause	Description
RQ1206_199	12.6.5.5.1	The parameter of GET_SSP_INFO command is defined as follows: On reception of the "Si3.GetSspInfo" function command, the Secondary Platform Bundle Loader shall: 2) Build aSspInfoPublic data structure defined in clause 12.6.2.2.2 of ETSI TS 103 666-2 [10] as follows: d) If GET_SSP_INFO command parameter is empty, the Secondary Platform Bundle Loader shall build aSspInfoPublic containing: • SspFamilyCryptoInfoBlock data structures as many as the number of family identifiers supported by the Secondary Platform Bundle Loader. Each SspFamilyCryptoInfoBlock data structure may contain aSspFamilyCryptoInfo. Each SspFamilyCrytoInfoBlock data structure may contain the set of SspOidCryptoInfoBlock data structures as many as custodians supported by the Secondary Platform Bundle Loader for the family identifier contained in the SspFamilyCryptoInfoBlock data structure. Each SspOidCryptoInfoBlock data structure shall contain the aCustodianOid and aSspOidCryptoInfo. The Secondary Platform Bundle Loader may include aSspGeneralCryptoInfo.
RQ1206_200	12.6.5.5.1	The parameter of GET_SSP_INFO command is defined as follows: On reception of the "Si3.GetSspInfo" function command, the Secondary Platform Bundle Loader shall: 3) Return ANY_OK with the aSspInfoPublic.
	12.6.5.5.2	***Si3.SetSpbmCredential***
RQ1206_201	12.6.5.5.2	The "Si3.SetSpbmCredential" function shall be used by the LBA during the bound SPB image download procedure as defined in clause 12.3.3.2 of ETSI TS 103 666-2 [10].
RQ1206_202	12.6.5.5.2	The LBA shall use "Si3.GetSspCredential" function to deliver aSpbmCredential to the Secondary Platform Bundle Loader.
RQ1206_203	12.6.5.5.2	The "Si3.SetSpbmCredential" function command shall be ANY_SET_PARAMETER command defined in ETSI TS 103 666-2 [10], clause 8.5.4 which allows the LBA to update the registry.
RQ1206_204	12.6.5.5.2	The parameter of ANY_SET_PARAMETER command shall contain the index of IDS_CREDENTIAL_PARAMETER registry and the aSpbmCredential data structure defined in clause 12.6.2.3 of ETSI TS 103 666-2 [10].
RQ1206_205	12.6.5.5.2	The LBA shall build the aSpbmCredential containing the aSpbFamilyId, the aSpbmKaCertificates, the aSspCiPkIdToBeUsed, and the aSspCryptoToBeUsed contained in the aSi2GetSpbmCertificateResponse.
RQ1206_206	12.6.5.5.2	On reception of the "Si3.SetSpbmCredential" command, the Secondary Platform Bundle Loader shall: 1) Set the received SpbmCredential data structure to the IDS_CREDENTIAL_PARAMETER registry.
RQ1206_207	12.6.5.5.2	On reception of the "Si3.SetSpbmCredential" command, the Secondary Platform Bundle Loader shall: 2) Verify the received elements as follows: a) Verify that the aSpbmKaCertificates contained in the aSpbmCredential based on the certification path verification as defined in clause 12.2.1.1.4 of ETSI TS 103 666-2 [10]. The trusted public key used to verify the aSpbmKaCertificates shall be allowed to be used for loading the Secondary Platform Bundles with the aSpbFamilyId and aCustodianOid contained in the aSpbmCredential. b) Verify that the aSspCiPkIdToBeUsed is supported by itself for the loading of the Secondary Platform Bundles with the aSpbFamilyId and the aCustodianOid contained in the aSpbmCredential. c) Verify that the aSspCryptoToBeUsed is supported by itself for the loading of the Secondary Platform Bundles with the aSpbFamilyId and the aCustodianOid contained in the aSpbmCredential.
RQ1206_208	12.6.5.5.2	On reception of the "Si3.SetSpbmCredential" command, the Secondary Platform Bundle Loader shall: 3) Select the appropriate Secondary Platform Bundle Loader certificate that shall be verifiable by the trusted public key which is indicated by the aSspCiPkIdToBeUsed contained in the aSpbmCredential.

Req.ID	Clause	Description
RQ1206_209	12.6.5.5.2	On reception of the "Si3.SetSpbmCredential" command, the Secondary Platform Bundle Loader shall: 4) Generate the following: a) A Secondary Platform Bundle Loader's ephemeral key pair. The domain parameter used to generate the ephemeral key pair shall be the same as the one indicated by the SubjectPublicKeyInfo in the SPB Manager certificate for key agreement contained in aSpbmCredential. b) ID_TRANSAC as defined in GlobalPlatform Open Firmware Loader for Tamper Resistant Element [13]. c) aTbsSspImageSessionToken as defined in clause 12.6.2.4 of ETSI TS 103 666-2 [10]. d) aSspImageSessionTokenSignature by signing the aTbsSspImageSessionTokenwith the private key of the SPB Manager corresponding to the SPB Manager certificate for digital signature. e) The first session key as defined in Global Platform Open Firmware Loader for Tamper Resistant Element [13]. The first session key shall be generated with the Secondary Platform Bundle Loader's ephemeral private key and the public key contained in the SPB Manager certificate for key agreement. NOTE: The first session key is the same as 'KS1' in GlobalPlatform Open Firmware Loader for Tamper Resistant Element [13].
RQ1206_210	12.6.5.5.2	On reception of the "Si3.SetSpbmCredential" command, the Secondary Platform Bundle Loader shall: 4) Generate the following: f) aSspInfoProtected as defined in clause 12.6.2.2.3 of ETSI TS 103 666-2 [10]. g) aTbsSspToken containing aCodeM, aChallengeS and aSspInfoProtected which shall be protected. The aCodeM and aChallengeS shall be the same as those in the aSpbmCredential contained in GET_SSP_CREDENTIAL command parameter. h) aTbsSspTokenSignature by signing the TbsSspToken with the private key of the SPB Manager corresponding to the SPB Manager certificate for digital signature. i) aM-SSP as defined in clause 12.6.2.4 of ETSI TS 103 666-2 [10]. The aM-SSP shall be generated by encrypting the aTbsSspToken, the aTbsSspTokenSignature, and the Secondary Platform Bundle Loader certificate for digital signature. The encryption algorithm indicated by the aSspCryptoToBeUsed and the first session key shall be used to generate the aM-SSP. j) aSspInfoProtected as defined in clause 12.6.2.2.3 of ETSI TS 103 666-2 [10].
RQ1206_211	12.6.5.5.2	On reception of the "Si3.SetSpbmCredential" command, the Secondary Platform Bundle Loader shall: 5) Generate aSspCredential as defined in clause 12.6.2.4 of ETSI TS 103 666-2 [10] and set the aSspCredential into the TRE_CREDENTIAL_PARAMETER registry.
RQ1206_212	12.6.5.5.2	On reception of the "Si3.SetSpbmCredential" command, the Secondary Platform Bundle Loader shall: 6) Return ANY_OK with the GetSspCredentialResponse data structure to the LBA.
	12.6.5.5.3	***Si3.LoadBoundSpbInfo***
RQ1206_213	12.6.5.5.3	The "Si3.LoadBoundSpbInfo" function shall be used by the LBA during the installation procedure as defined in clause 12.3.4 of ETSI TS 103 666-2 [10].
RQ1206_214	12.6.5.5.3	The LBA shall use the "Si3.LoadBoundSpbInfo" function to provide the Secondary Platform Bundle Loader with the aDoOperateParameter contained in the bound SPB image received from the SPB Manager as the response of the "Si2.GetBoundSpbImage" function.
RQ1206_215	12.6.5.5.3	The "Si3.LoadBoundSpbInfo" function command shall be OFL_DO_OPERATE as defined in GlobalPlatform VPP - OFL VNP Extension [16].
RQ1206_216	12.6.5.5.3	The parameter of OFL_DO_OPERATE command shall be aDoOperateParameter defined in clause 12.6.2.5 of ETSI TS 103 666-2 [10].
RQ1206_217	12.6.5.5.3	On reception of the "Si3.LoadBoundSpbInfo" function command, the Secondary Platform Bundle Loader shall: 1) Verify the aSpbmCerts contained in the aDoOperateParameter based on the certification path verification as defined in clause 12.2.1.1.4 of ETSI TS 103 666-2 [10]. The trusted public key used to verify the aSpbmCerts shall be the same as the one used to verify the aSpbmKaCertificates.
RQ1206_218	12.6.5.5.3	On reception of the "Si3.LoadBoundSpbInfo" function command, the Secondary Platform Bundle Loader shall: 2) Verify the aSpbmTokenSignature contained in the aSpbmToken by using the SPB Manager certificate for digital signature. The SPB Manager certificate for digital signature shall be the last certificate in the aSpbmCerts.

Req.ID	Clause	Description
RQ1206_219	12.6.5.5.3	On reception of the "Si3.LoadBoundSpbInfo" function command, the Secondary Platform Bundle Loader shall: 3) Verify that the aIdTransac contained in the aTbsSpbmToken matches to the previously generated ID_TRANSAC.
RQ1206_220	12.6.5.5.3	On reception of the "Si3.LoadBoundSpbInfo" function command, the Secondary Platform Bundle Loader shall: 4) Generate the second session key as defined in GlobalPlatform Open Firmware Loader for Tamper Resistant Element [13]. The second session key shall be generated with the aEPkSpbmKa contained in the aTbsSpbmToken and the Secondary Platform Bundle Loader's ephemeral private key. NOTE: The second session key is the same as 'KS2' in GlobalPlatform Open Firmware Loader for Tamper Resistant Element [13].
RQ1206_221	12.6.5.5.3	On reception of the "Si3.LoadBoundSpbInfo" function command, the Secondary Platform Bundle Loader shall: 5) Obtain the TIME_STAMP by decrypting the aM-TimeStamp by using the first session key and the encryption algorithm indicated by the aEncryptionType as defined in GlobalPlatform Open Firmware Loader for Tamper Resistant Element [13].
RQ1206_222	12.6.5.5.3	On reception of the "Si3.LoadBoundSpbInfo" function command, the Secondary Platform Bundle Loader shall: 6) Obtain the ATK.ARP.DS by decrypting the aM-ARP by using the second session key and the encryption algorithm indicated by the aEncryptionType as defined in GlobalPlatform Open Firmware Loader for Tamper Resistant Element [13].
RQ1206_223	12.6.5.5.3	On reception of the "Si3.LoadBoundSpbInfo" function command, the Secondary Platform Bundle Loader shall: 7) Obtain Image Descriptor by decrypting the aM-IMD by using the second session key and the encryption algorithm indicated by the aEncryptionType as defined in GlobalPlatform Open Firmware Loader for Tamper Resistant Element [13].
RQ1206_224	12.6.5.5.3	On reception of the "Si3.LoadBoundSpbInfo" function command, the Secondary Platform Bundle Loader shall: 8) Verify the Image Descriptor as follows: a) Verify that the family identifier contained in the Image Descriptor matches to the value of the family identifier in SSP_INFO_PUBLIC registry.
RQ1206_225	12.6.5.5.3	On reception of the "Si3.LoadBoundSpbInfo" function command, the Secondary Platform Bundle Loader shall: 9) Verify the aSpbMetadata as follows: a) Verify that the family identifier contained in the aSpbMetadata matches to the value of the family identifier in SSP_INFO_PUBLIC registry. b) Verify that the aSpbId contained in the aSpbMetadata matches to the value of the public UUID of the image contained in the Image Descriptor. c) Verify the aFamilySpecificData and aOidSpecificMetadata (out of scope of the present document).
RQ1206_226	12.6.5.5.3	On reception of the "Si3.LoadBoundSpbInfo" function command, the Secondary Platform Bundle Loader shall: 10) Verify that the trusted public key used to verify the SPB Manager certificate is one of the trusted public keys supported by the Secondary Platform Bundle Loader used to load the Secondary Platform Bundle according to the rules below: a) if the Secondary Platform Bundle family identifier is not part of any SspFamilyCryptoInfoBlock: • the keys in aSspGeneralCryptoInfo. b) Else: • the keys in aSspFamilyCryptoInfo, if none of the custodian OIDs in aSpbMetadata (either aCustodianOid or in aSupportedCustodianList) is part of any aSspOidCryptoInfoBlock; • else, the keys in aSspOidCryptoInfo of the SspOidCryptoInfoBlock data structure which the custodian OID has been found in aSpbMetadata.
RQ1206_226a	12.6.5.5.3	On reception of the "Si3.LoadBoundSpbInfo" function command, the Secondary Platform Bundle Loader shall: 10) Verify that the trusted public key used to verify the SPB Manager certificate is one of the trusted public keys supported by the Secondary Platform Bundle Loader used to load the Secondary Platform Bundle, if the Secondary Platform Bundle family identifier is not part of any SspFamilyCryptoInfoBlock: • the keys in aSspGeneralCryptoInfo.

Req.ID	Clause	Description
RQ1206_226b	12.6.5.5.3	On reception of the "Si3.LoadBoundSpbInfo" function command, the Secondary Platform Bundle Loader shall: 10) Verify that the trusted public key used to verify the SPB Manager certificate is one of the trusted public keys supported by the Secondary Platform Bundle Loader used to load the Secondary Platform Bundle if the Secondary Platform Bundle family identifier is part of any SspFamilyCryptoInfoBlock: • the keys in aSspFamilyCryptoInfo, if none of the custodian OIDs in aSpbMetadata (either aCustodianOid or in aSupportedCustodianList) is part of any aSspOidCryptoInfoBlock; • else, the keys in aSspOidCryptoInfo of the SspOidCryptoInfoBlock data structure which the custodian OID has been found in aSpbMetadata.
RQ1206_227		Void
RQ1206_228		Void
RQ1206_229	12.6.5.5.3	On reception of the "Si3.LoadBoundSpbInfo" function command, the Secondary Platform Bundle Loader shall: 11) Return ANY_OK without any parameters to the LBA.
	12.6.5.5.4	***Si3.LoadBoundSpbSds***
RQ1206_230	12.6.5.5.4	The "Si3.LoadBoundSpbSds" function shall be used by the LBA during the installation procedure as defined in clause 12.3.4 of ETSI TS 103 666-2 [10].
RQ1206_231	12.6.5.5.4	The LBA shall use the "Si3.LoadBoundSpbSds" function to provide the Secondary Platform Bundle Loader with an element of aChangeSegmentParameter contained in the bound SPB image received from the SPB Manager as the response of the "Si2.GetBoundSpbImage" function.
RQ1206_232	12.6.5.5.4	The "Si3.LoadBoundSpbSds" function command shall be OFL_CHANGE_SEGMENT as defined in GlobalPlatform VPP - OFL VNP Extension [16].
RQ1206_233	12.6.5.5.4	The parameter of OFL_CHANGE_SEGMENT command shall be aChangeSegmentParameter defined in clause 12.6.2.5 of ETSI TS 103 666-2 [10].
RQ1206_234	12.6.5.5.4	On reception of the "Si3.LoadBoundSpbSds" function command, the Secondary Platform Bundle Loader shall decrypt aChangeSegmentParameter to obtain Segment Descriptor as defined in GlobalPlatform Open Firmware Loader for Tamper Resistant Element [13]. The Secondary Platform Bundle Loader shall return ANY_OK to the LBA after successful decryption of the aChangeSegmentParameter. NOTE: The aChangeSegmentParameter is the same as the Segment Descriptor Structure defined in GlobalPlatform Open Firmware Loader for Tamper Resistant Element [13].
	12.6.5.5.5	***Si3.LoadBoundSpbSeg***
RQ1206_235	12.6.5.5.5	The "Si3.LoadBoundSpbSeg" function shall be used by the LBA during the installation procedure as defined in clause 12.3.4 of ETSI TS 103 666-2 [10].
RQ1206_236	12.6.5.5.5	The LBA shall use the "Si3.LoadBoundSpbSeg" function to provide the Secondary Platform Bundle Loader with an element of aLoadSegmentParameter contained in the bound SPB image received from the SPB Manager as the response of the "Si2.GetBoundSpbImage" function.
RQ1206_237	12.6.5.5.5	The "Si3.LoadBoundSpbSeg" function command shall be OFL_LOAD_SEGMENT as defined in GlobalPlatform VPP - OFL VNP Extension [16].
RQ1206_238	12.6.5.5.5	The parameter of OFL_LOAD_SEGMENT command shall be aLoadSegmentParameter defined in clause 12.6.2.5 of ETSI TS 103 666-2 [10].
RQ1206_239	12.6.5.5.5	On reception of the "Si3.LoadBoundSpbSeg" function command, the Secondary Platform Bundle Loader shall decrypt the aLoadSegmentParameter and install the decrypted segment into the iSSP as defined in GlobalPlatform Open Firmware Loader for Tamper Resistant Element [13]. The Secondary Platform Bundle Loader shall return ANY_OK to the LBA after the successful installation of the segment. NOTE: The aLoadSegmentParameter is the same as the Segment Structure defined in GlobalPlatform Open Firmware Loader for Tamper Resistant Element [13].
	12.6.5.5.6	***Si3.GetSspCredential***
RQ1206_240	12.6.5.5.6	The "Si3.GetSspCredential" function shall be used by the LBA during the bound SPB image download procedure as defined in clause 12.3.3.2 of ETSI TS 103 666-2 [10].
RQ1206_241	12.6.5.5.6	The LBA shall use the "Si3.GetSspCredential" function to retrieve aSspCredential from the Secondary Platform Bundle Loader.
RQ1206_242	12.6.5.5.6	The "Si3.GetSspCredential" function command shall be ANY_GET_PARAMETER command defined in ETSI TS 103 666-2 [10], clause 8.5.4 which allows the LBA to retrieve the value of the registry.
RQ1206_243	12.6.5.5.6	The parameter of ANY_GET_PARAMETER command shall contain the index of TRE_CREDENTIAL_PARAMETER registry.
RQ1206_244	12.6.5.5.6	On reception of the "Si3.GetSspCredential" command, the Secondary Platform Bundle Loader shall return ANY_OK with the value of TRE_CREDENTIAL_PARAMETER registry which contains aSspCredential.

Req.ID	Clause	Description
	12.6.5.5.7	***Si3.EnableSpb***
RQ1206_245	12.6.5.5.7	The "Si3.EnableSpb" function shall be used by the LBA for the procedure to enable a Secondary Platform Bundle as defined in clause 12.4.1 of ETSI TS 103 666-2 [10].
RQ1206_246	12.6.5.5.7	The LBA shall use the "Si3.EnableSpb" function to provide the Secondary Platform Bundle Loader with the Secondary Platform Bundle identifier to enable.
RQ1206_247	12.6.5.5.7	The "Si3.EnableSpb" function command shall be OFL_ENABLE_FIRMWARE as defined in GlobalPlatform VPP - OFL VNP Extension [16].
RQ1206_248	12.6.5.5.7	The Secondary Platform Bundle identifier to enable shall be the Public Image UUID as defined in GlobalPlatform VPP - OFL VNP Extension [16].
RQ1206_249	12.6.5.5.7	If the Secondary Platform Bundle to enable is a Telecom Secondary Platform Bundle, the Secondary Platform Bundle Loader shall check the number of currently enabled Telecom Secondary Platform Bundles and operate depending on the value of TELECOM_CAPABILITY as follows: • If the number of currently enabled Telecom Secondary Platform Bundles is smaller than the value of TELECOM_CAPABILITY, the Secondary Platform Bundle Loader shall enable the Telecom Secondary Platform Bundle to enable. • Otherwise, the Secondary Platform Bundle Loader shall reject the "Si3.EnableSpb" command with an error indicating that enabling the Telecom Secondary Platform Bundle is limited by TELECOM_CAPABILITY.
RQ1206_250	12.6.5.5.7	After successfully enabling the Secondary Platform Bundle, the Secondary Platform Bundle Loader shall update the value of the state to 'Enabled' in the Firmware session of that Secondary Platform Bundle.
	12.6.5.5.8	***Si3.DisableSpb***
RQ1206_251	12.6.5.5.8	The "Si3.DisableSpb" function shall be used by the LBA for the procedure to disable a Secondary Platform Bundle as defined in clause 12.4.2 of ETSI TS 103 666-2 [10].
RQ1206_252	12.6.5.5.8	The LBA shall use the "Si3.DisableSpb" function to provide the Secondary Platform Bundle Loader with the Secondary Platform Bundle identifier to disable.
RQ1206_253	12.6.5.5.8	The "Si3.DisableSpb" function command shall be OFL_DISABLE_FIRMWARE as defined in GlobalPlatform VPP - OFL VNP Extension [11].
RQ1206_254	12.6.5.5.8	The Secondary Platform Bundle identifier to disable shall be the Public Image UUID as defined in GlobalPlatform VPP - OFL VNP Extension [16].
RQ1206_255	12.6.5.5.8	After successfully disabling the Secondary Platform Bundle, the Secondary Platform Bundle Loader shall update the value of the state to 'Disabled' in the Firmware session of that Secondary Platform Bundle.
	12.6.5.5.9	***Si3.DeleteSpb***
RQ1206_256	12.6.5.5.9	The "Si3.DeleteSpb" function shall be used by the LBA for the procedure to delete a Secondary Platform Bundle as defined in clause 12.4.3 of ETSI TS 103 666-2 [10].
RQ1206_257	12.6.5.5.9	The LBA shall use the "Si3.DeleteSpb" function to provide the Secondary Platform Bundle Loader with the Secondary Platform Bundle identifier to delete.
RQ1206_258	12.6.5.5.9	The "Si3.DeleteSpb" function command shall be OFL_DELETE_SESSION as defined in GlobalPlatform VPP - OFL VNP Extension [16].
RQ1206_259	12.6.5.5.9	The Secondary Platform Bundle identifier to disable shall be the Public Image UUID as defined in GlobalPlatform VPP - OFL VNP Extension [16].
	12.6.5.5.10	***Si3.GetSpbMetadata***
RQ1206_260	12.6.5.5.10	The "Si3.GetSpbMetadata" function shall be used by the LBA to retrieve the SPB metadata of a Secondary Platform Bundle container installed in the iSSP.
RQ1206_261	12.6.5.5.10	The "Si3.GetSpbMetadata" function command shall be GET_SPB_METADATA.
RQ1206_262	12.6.5.5.10	The parameter of GET_SPB_METADATA command is a Secondary Platform Bundle identifier.
RQ1206_263	12.6.5.5.10	The Secondary Platform Bundle identifier shall be the Public Image UUID as defined in GlobalPlatform VPP - OFL VNP Extension [16].
RQ1206_264	12.6.5.5.10	On reception of the "Si3.GetSpbMetadata" function command, the Secondary Platform Bundle Loader shall: 1) find the firmware session which contains the Public Image UUID same as the received Secondary Platform Bundle identifier; 2) extract the SPB metadata contained in that firmware session; 3) return ANY_OK with the SPB metadata as the "Si3.GetSpbMetadata" function response.
	12.6.5.5.11	***Si3.UpdateSpbState***
RQ1206_265	12.6.5.5.11	The "Si3.UpdateSpbState" function shall be used by the LBA during the SPB state retrieving procedure as defined in clause 12.4.5 of ETSI TS 103 666-2 [10].
RQ1206_266	12.6.5.5.11	The LBA shall use "Si3.UpdatetSpbState" function to request the Secondary Platform Bundle Loader to update the value of SPB_ID registry.

Req.ID	Clause	Description
RQ1206_267	12.6.5.5.11	The "Si3.UpdateSpbState" function command shall be ANY_SET_PARAMETER command defined in ETSI TS 103 666-1 [9], clause 8.5.4 which allows the LBA to update the registry.
RQ1206_268	12.6.5.5.11	The parameter of ANY_SET_PARAMTER command shall contain the index of SPB_ID registry and the Secondary Platform Bundle identifier.
RQ1206_269	12.6.5.5.11	On reception of the "Si3.UpdateSpbState" command, the Secondary Platform Bundle Loader shall: 1) Set the received Secondary Platform Bundle identifier (SpbId) to the SPB_ID registry. 2) Extract the SPB state from the firmware session which contains the Public Image UUID same as the received SpbId. 3) Update the SPB_STATE registry with the value of the extracted SPB state. 4) Return ANY_OK to the LBA as "Si3.UpdateSpbState" function response to the LBA.
	12.6.5.5.12	***Si3.GetSpbState***
RQ1206_270	12.6.5.5.12	The "Si3.GetSpbState" function shall be used by the LBA during the SPB state retrieving procedure as defined in clause 12.4.5 of ETSI TS 103 666-2 [10].
RQ1206_271	12.6.5.5.12	The "Si3.GetSpbState" function command shall be ANY_GET_PARAMETER command defined in ETSI TS 103 666-1 [9], clause 8.5.4 which allows the LBA to retrieve the value of the registry
RQ1206_272	12.6.5.5.12	The parameter of ANY_GET_PARAMETER command shall contain the index of SPB_STATE registry.
RQ1206_273	12.6.5.5.12	On reception of the "Si3.GetSpbState" command, the Secondary Platform Bundle Loader shall return ANY_OK with the value of SPB_STATE registry to the LBA.

5.11 Requirements not covered by ETSI test descriptions

5.11.1 Requirements assigned to the Security Certification labs

As mentioned in clause 4.1.1 of the present document, Evaluation Level Assurance certification for the SSP Primary Platform and the SPB certification by composition on the Primary Platform, except for the SPB loader, is out of scope of the present document. Requirements the iSSP maker identifies to be fulfilled for the intended EAL best will be provided to a security certification lab, accredited by the certification body, where the verification can take place.

Therefore, the following requirements will not be verified by tests defined in the present document:

RQ number	RQ number	RQ number	RQ number	RQ number	RQ number
RQ0502_002	RQ0701_010	RQ0701_017	RQ0701_021	RQ0702_007	RQ1101_001
RQ0502_003	RQ0701_011	RQ0701_018	RQ0701_023	RQ0703_002	RQ1102_001
RQ0701_008	RQ0701_013	RQ0701_019	RQ0701_025	RQ0803_001	RQ1102_002
RQ0701_009	RQ0701_014	RQ0701_020	RQ0701_026	RQ1002_002	

5.11.2 Requirements referencing GlobalPlatform specifications

Some requirements identified in ETSI TS 103 666-2 [10] are based on descriptions or specifications generated by GlobalPlatform. Services, interfaces and functionality described by GlobalPlatform specifications need to fulfill GlobalPlatform regulations.

Therefore, the following requirements will not be verified by tests defined in the present document:

RQ number	RQ number	RQ number	RQ number	RQ number	RQ number
RQ0601_001	RQ0701_001	RQ0701_007	RQ0702_002	RQ0703_008	RQ0902_001
RQ0602_001	RQ0701_002	RQ0701_015	RQ0702_003	RQ0703_010	RQ0902_002
RQ0603_001	RQ0701_003	RQ0701_016	RQ0702_004	RQ0703_015	RQ0903_001
RQ0604_001	RQ0701_004	RQ0701_022	RQ0702_005	RQ0706_001	RQ0904_001
RQ0605_001	RQ0701_005	RQ0701_028	RQ0702_006	RQ0801_001	
RQ0606_001	RQ0701_006	RQ0702_001	RQ0703_006	RQ0802_001	

5.11.3 Descriptive requirements and not explicitly testable requirements

Some requirements identified in ETSI TS 103 666-2 [10] are descriptive text. In some cases, it is not possible to explicitly verify requirements generated from descriptive text in other cases the verification of such requirements is out of scope of the present document. Implicitly verified and 'out of scope' requirements are identified and listed in the respective clauses.

6 Security requirements and iSSP architecture testing

6.1 Configurations

There are no specific configurations defined for security requirements an iSSP architecture testing.

6.2 Procedures

There are no specific procedures defined for security requirements an iSSP architecture testing.

6.3 Test descriptions

There are no specific test descriptions defined for testing the Security requirements and iSSP architecture.

6.4 Requirements verified elsewhere

6.4.1 Overview - Security requirements

The following requirements, identified in ETSI TS 103 666-2 [10] clause 5.2 are not tested in accordance with the present document, as they are either referencing requirements from other standardization bodies; best verified by a certified security certification laboratory outside the ETSI remits, or as they are descriptive without identifiable specific usage:

RQ0502_001, RQ0502_002, RQ0502_003, RQ0502_004.

6.4.2 iSSP Architecture

The following requirements, identified in ETSI TS 103 666-2 [10] clause 6 are not tested in accordance with the present document, as they are referencing requirements from another standardization body (GlobalPlatform):

RQ0601_001, RQ0602_001, RQ0603_001, RQ0604_001, RQ0605_001, RQ0606_001.

7 Primary Platform

7.1 Hardware Platform

7.1.1 Configurations

There are no specific configurations defined for hardware platform testing.

7.1.2 Procedures

There are no specific procedures defined for hardware platform testing.

7.1.3 Test descriptions

There are no specific test descriptions defined for testing the Low-level Operating System.

7.1.4 Requirements not testable, implicitly verified or verified elsewhere

7.1.4.1 Architecture

The following requirements, identified in ETSI TS 103 666-2 [10] are not tested in accordance with the present document, as they are referencing requirements from another standardization body (GlobalPlatform):

RQ0701_001, RQ0701_002, RQ0701_003, RQ0701_004.

7.1.4.2 Security functions

The following requirements, identified in ETSI TS 103 666-2 [10] are not tested in accordance with the present document, as they are referencing requirements from another standardization body (GlobalPlatform):

RQ0701_005, RQ0701_006, RQ0701_007, RQ0701_015, RQ0701_016.

The following requirements, identified in ETSI TS 103 666-2 [10] are not tested in accordance with the present document:

RQ0701_008, RQ0701_009, RQ0701_010, RQ0701_011, RQ0701_012, RQ0701_013, RQ0701_014, RQ0701_017, RQ0701_018.

7.1.4.3 Memories

The following requirements, identified in ETSI TS 103 666-2 [10] are not tested in accordance with the present document, as they need to be verified by a security certification laboratory outside the ETSI domain (in accordance with BSI regulations):

RQ0701_019, RQ0701_020, RQ0701_021.

7.1.4.4 Cryptographic functions

The following requirement, identified in ETSI TS 103 666-2 [10] is not tested in accordance with the present document, as it is referencing a requirement from another standardization body (GlobalPlatform):

RQ0701_0022.

7.1.4.5 Clock

The following requirement, identified in ETSI TS 103 666-2 [10] is not tested in accordance with the present document, as it needs to be verified by a security certification laboratory outside the ETSI domain (in accordance with BSI regulations):

RQ0701_023.

The following requirement, identified in ETSI TS 103 666-2 [10] refers to descriptive text in ETSI TS 103 666-1 [9]. It shall be verified in accordance with tests defined for the clock signal in ETSI TS 103 999-1 [11] clause 6.3 (see note):

RQ0701_024.

NOTE: Check with the recent version of ETSI TS 103 999-1 [11] if appropriate tests are defined.

7.1.4.6 SSP internal interconnect

The following requirement, identified in ETSI TS 103 666-2 [10] is not tested in accordance with the present document, as it needs to be verified by a security certification laboratory outside the ETSI domain (in accordance with BSI regulations):

RQ0701_025.

7.1.4.7 Secure CPU

The following requirement, identified in ETSI TS 103 666-2 [10] is not tested in accordance with the present document, as it is recommended to be verified by a security certification laboratory outside the ETSI remits (e.g.: in accordance with BSI regulations):

RQ0701_026.

The following requirements, identified in ETSI TS 103 666-2 [10] are not tested in accordance with the present document, as they are referencing requirements from another standardization body (GlobalPlatform):

RQ0701_027, RQ0701_028.

7.1.4.8 Random Number Generator

The following requirement, identified in ETSI TS 103 666-2 [10] is not tested in accordance with the present document, as it is referencing a requirement from another standardization body (GlobalPlatform):

RQ0701_029.

7.2 Low-level Operating System

7.2.1 Configurations

There are no specific configurations defined for low-level operating system testing.

7.2.2 Procedures

There are no specific procedures defined for low-level operating system testing.

7.2.3 Test descriptions

There are no specific test descriptions defined for testing the Low-level Operating System.

7.2.4 Requirements not testable, implicitly verified or verified elsewhere

7.2.4.1 Introductions

The following requirement, identified in ETSI TS 103 666-2 [10] is not tested in accordance with the present document, as it is referencing a requirement from another standardization body (GlobalPlatform):

RQ0702_001.

7.2.4.2 Kernel objects

The following requirement, identified in ETSI TS 103 666-2 [10] is not tested in accordance with the present document, as it is referencing a requirement from another standardization body (GlobalPlatform):

RQ0702_002.

7.2.4.3 Global requirements and mandatory Access Control rules

The following requirement, identified in ETSI TS 103 666-2 [10] is not tested in accordance with the present document, as it is referencing a requirement from another standardization body (GlobalPlatform):

RQ0702_003.

7.2.4.4 Process states diagram

The following requirement, identified in ETSI TS 103 666-2 [10] is not tested in accordance with the present document, as it is referencing a requirement from another standardization body (GlobalPlatform):

RQ0702_004.

7.2.4.5 Definition of the process states

The following requirement, identified in ETSI TS 103 666-2 [10] is not tested in accordance with the present document, as it is referencing a requirement from another standardization body (GlobalPlatform):

RQ0702_005.

7.2.4.6 Mandatory access control

The following requirement, identified in ETSI TS 103 666-2 [10] is not tested in accordance with the present document, as it is referencing a requirement from another standardization body (GlobalPlatform):

RQ0702_006.

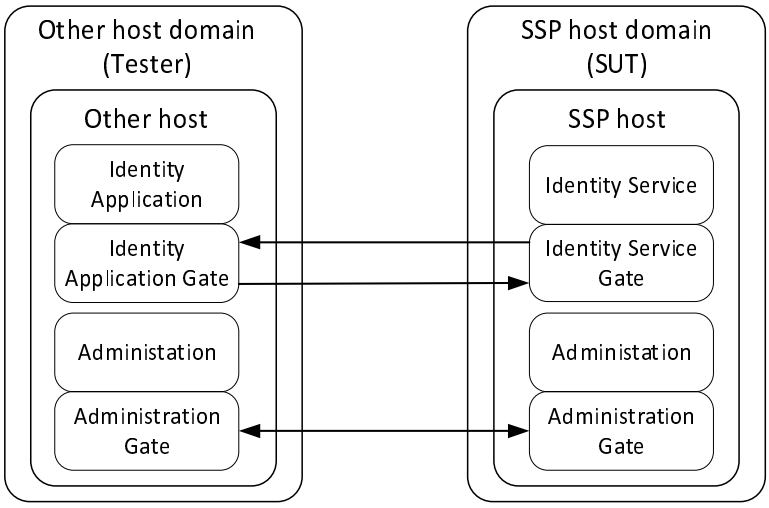
The following requirement, identified in ETSI TS 103 666-2 [10] is not tested in accordance with the present document:

RQ0702_007.

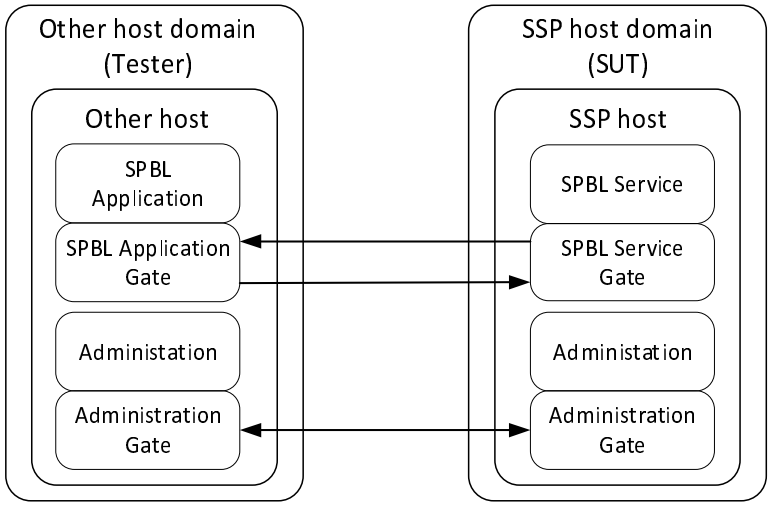
7.3 Services

7.3.1 Configurations

7.3.1.1 CSVC_311

Configuration ID	CSVC_311
Configuration description	

7.3.1.2 CSVC_312

Configuration ID	CSVC_312
Configuration description	

7.3.2 Procedures

7.3.2.1 PSVC_321 - Open a pipe session on the Identity Service Gate

Procedure ID	PSVC_321
Objectives	The SSP host shall have implemented the registry entries of the OFL service gate defined in GlobalPlatform OFL VNP Extension [16].
Configuration reference	CSVC_311
Initial conditions	
Test 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).

7.3.2.1 PSVC_322 - Open a pipe session on the SPBL Service Gate

Procedure ID	PSVC_322
Objectives	The other host shall be able to open a pipe session to the service gate of the SSP host. The SPBL service identifier is defined as the OFL service identifier in GlobalPlatform OFL VNP extension [16].
Configuration reference	CSVC_312
Initial conditions	
Test 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 SPBL service gate. • $GATE_{SPBL}$: The UUID gate identifier of the SPBL service gate (BB780E30-419A-5B71-9B98-18A042E75899).
2	Administration gate sends EVT_ADM_BIND to Administration gate in the other host with: • $PIPE_{YX}$: a dynamically assigned pipe identifier for the SPBL application gate. • $GATE_{SPBL}$: The UUID gate identifier of the identity gate (BB780E30-419A-5B71-9B98-18A042E75899).
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 'BB780E30-419A-5B71-9B98-18A042E75899' shall be present.

7.3.3 Test descriptions

7.3.3.1 Secondary Platform Bundle Loader

7.3.3.1.1 SVC_3311 - SPBL ARP state

Test ID	SVC_3311	
Test objectives	To verify the availability and correct configuration of the ARP state from the extract of the OFL_DO_OPERATE command.	
Configuration reference	CSVC_311, CSVC_312	
Initial conditions		
Pipe sessions are opened and established as defined in PSVC_321 and PSVC_322.		
Test sequence		
Step	**Description**	**Requirements**
1	The SPBL application gate sends the SET_PARAMETER command for storing the SPBM credentials parameters from the Service Provider in the SPBL service gate.	
2	The SPBL service gate returns ANY_OK if the command is successfully executed.	
3	The SPBL application gate sends the GET_PARAMETER (CODE_M ?) command for getting the initial SSP credentials parameter computed from the Service Provider parameter.	
4	The SPBL service gate returns ANY_OK if the command is successfully executed.	
5	The SPBL application gate sends the OFL_DO_OPERATE command with M_3 encrypted by $KS2_1^3$ including ARP '02' to administrate the SSP (ARP management) to the SPBL host.	
6	The SPBL service gate returns ANY_OK if the command is successfully executed.	RQ0703_001

7.3.3.1.2 SVC_3312 - Registry entries in the SPBL Service Gate

Test ID	SVC_3312	
Test objectives	To verify that the SPBL has implemented at least the registry entries provided in Table 7.1 of ETSI TS 103 666-2 [10].	
Configuration reference	CSVC_311, CSVC_312	
Initial conditions		
Pipe sessions are opened and established as defined in PSVC_321 and PSVC_322.		
Test sequence		
Step	**Description**	**Requirements**
1	The SPBL application gate sends the SET_PARAMETER command for storing the SPBM credentials parameters from the Service Provider in the SPBL service gate.	
2	The SPBL service gate returns ANY_OK if the command is successfully executed.	
3	The SPBL application gate sends the GET_PARAMETER (CODE_M) command for getting the initial SSP credentials parameter computed from the Service Provider parameter.	
4	The SPBL service gate returns ANY_OK if the command is successfully executed.	
5	The SPBL application gate sends the OFL_DO_OPERATE command with the registry entries defined in Table 7.1 of ETSI TS 103 666-2 [10] to the SPBL host.	
6	The SPBL service gate returns ANY_OK if the command is successfully executed.	RQ0703_003

7.3.3.1.3 SVC_3313 - Additional registry entries in the SPBL Service Gate

Test ID	SVC_3313	
Test objectives	To verify that the SPBL has implemented the additional registry entries provided in Table 7.2 of ETSI TS 103 666-2 [10].	
Configuration reference	CSVC_311, CSVC_312	
Initial conditions		
Pipe sessions are opened and established as defined in PSVC_321 and PSVC_322.		
Test sequence		
Step	**Description**	**Requirements**
1	The SPBL application gate sends the SET_PARAMETER command for storing the SPBM credentials parameters from the Service Provider in the SPBL service gate.	
2	The SPBL service gate returns ANY_OK if the command is successfully executed.	
3	The SPBL application gate sends the GET_PARAMETER (CODE_M ?) command for getting the initial SSP credentials parameter computed from the Service Provider parameter.	
4	The SPBL service gate returns ANY_OK if the command is successfully executed.	
5	The SPBL application gate sends the OFL_DO_OPERATE command with mandatory registry entries defined in Table 7.1 and additional registry entries defined in Table 7.2 of ETSI TS 103 666-2 [10] to the SPBL host.	
6	The SPBL service gate returns ANY_OK if the command is successfully executed.	RQ0703_004

7.3.3.1.4 SVC_3314 – Content of registry entry TELECOM_CAPABILITY

Test ID	SVC_3314	
Test objectives	To verify that the registry entry TELECOM_CAPABILITY on an iSSP hosting a Telecom Secondary Platform Bundle contains the maximum number of distinct concurrent 3GPP network registrations.	
Configuration reference	CSVC_311, CSVC_312	
Initial conditions		
Pipe sessions are opened and established as defined in PSVC_321 and PSVC_322. The pre-configured SPB container indicates a maximum number of concurrent 3GPP network registrations other that the default.		
Test sequence		
Step	**Description**	**Requirements**
1	The SPBL application gate sends the SET_PARAMETER command for storing the SPBM credentials parameters from the Service Provider in the SPBL service gate.	
2	The SPBL service gate returns ANY_OK if the command is successfully executed.	

3	The SPBL application gate sends the GET_PARAMETER (CODE_M) command for getting the initial SSP credentials parameter computed from the Service Provider parameter.	
4	The SPBL service gate returns ANY_OK if the command is successfully executed.	
5	The SPBL application gate sends the OFL_DO_OPERATE command containing the conditional parameter TELECOM_CAPABILITY to the SPBL host.	
6	The SPBL service gate returns ANY_OK if the command is successfully executed.	RQ0703_005

7.3.3.1.5 SVC_3315 – Additional responses supported by the OFL Service Gate #1

Test ID	SVC_3315	
Test objectives	To verify that the SPBL supports the additional response: eSPBL_E_NO_CI_FOR_SPBM_VERIFICATION as defined in Table 7.4 to the additional command entry: GET_SSP_INFO_PARAMETER defined in Table 7.3 of ETSI TS 103 666-2 [10].	
Configuration reference	CSVC_311, CSVC_312	
Initial conditions		
Pipe sessions are opened and established as defined in PSVC_321 and PSVC_322. The pre-configured SPB container indicates that the provided CIs are not supported for SPBM verification.		
Test sequence		
Step	**Description**	**Requirements**
1	The SPBL application gate sends the SET_PARAMETER command for storing the SPBM credentials parameters from the Service Provider in the SPBL service gate.	
2	The SPBL service gate returns ANY_OK if the command is successfully executed.	
3	The SPBL application gate sends the GET_SSP_INFO_PARAMETER command.	
4	The SPBL service gate sends the response value '10' (eSPBL_E_NO_CI_FOR_SPBM_VERIFICATION).	RQ0703_009

7.3.3.1.6 SVC_3316 - Additional responses supported by the OFL Service Gate #2

Test ID	SVC_3316	
Test objectives	To verify that the SPBL supports the additional response: eSPBL_E_NO_CI_FOR_SPBL_VERIFICATION as defined in Table 7.4 to the additional command entry: GET_SSP_INFO_PARAMETER defined in Table 7.3 of ETSI TS 103 666-2 [10].	
Configuration reference	CSVC_311, CSVC_312	
Initial conditions		
Pipe sessions are opened and established as defined in PSVC_321 and PSVC_322. The pre-configured SPB container indicates that it does not support CIs to sign the SPBL.		
Test sequence		
Step	**Description**	**Requirements**
1	The SPBL application gate sends the SET_PARAMETER command for storing the SPBM credentials parameters from the Service Provider in the SPBL service gate.	
2	The SPBL service gate returns ANY_OK if the command is successfully executed.	
3	The SPBL application gate sends the GET_SSP_INFO_PARAMETER command.	
4	The SPBL service gate sends the response value '11' (eSPBL_E_NO_CI_FOR_SPBL_VERIFICATION).	RQ0703_010

7.3.3.1.7 SVC_3317 - Additional responses supported by the OFL Service Gate #3

Test ID	SVC_3317
Test objectives	To verify that the SPBL supports the additional response: eSPBL_E_NO_CI_FOR_KEYAGREEMENT as defined in Table 7.4 to the additional command entry: GET_SSP_INFO_PARAMETER defined in Table 7.3 of ETSI TS 103 666-2 [10].
Configuration reference	CSVC_311, CSVC_312
Initial conditions	
Pipe sessions are opened and established as defined in PSVC_321 and PSVC_322. The pre-configured SPB container indicates that it does not support any CIs for key agreement.	

Test sequence		
Step	**Description**	**Requirements**
1	The SPBL application gate sends the SET_PARAMETER command for storing the SPBM credentials parameters from the Service Provider in the SPBL service gate.	
2	The SPBL service gate returns ANY_OK if the command is successfully executed.	
3	The SPBL application gate sends the GET_SSP_INFO_PARAMETER command.	
4	The SPBL service gate sends the response value '12' (eSPBL_E_NO_CI_FOR_KEYAGREEMENT).	RQ0703_011

7.3.3.1.8 SVC_3318 - Additional responses supported by the OFL Service Gate #4

Test ID	SVC_3318
Test objectives	To verify that the SPBL supports the additional response: eSPBL_E_NO_SUPPORTED_CRYPTO as defined in Table 7.4 to the additional command entry: GET_SSP_INFO_PARAMETER defined in Table 7.3 of ETSI TS 103 666-2 [10].
Configuration reference	CSVC_311, CSVC_312
Initial conditions	
Pipe sessions are opened and established as defined in PSVC_321 and PSVC_322. The pre-configured SPB container indicates that it does not support any cryptographic algorithms.	
Test sequence	
Step	**Description**	**Requirements**
1	The SPBL application gate sends the SET_PARAMETER command for storing the SPBM credentials parameters from the Service Provider in the SPBL service gate.	
2	The SPBL service gate returns ANY_OK if the command is successfully executed.	
3	The SPBL application gate sends the GET_SSP_INFO_PARAMETER command.	
4	The SPBL service gate sends the response value '13' (eSPBL_E_NO_SUPPORTED_CRYPTO).	RQ0703_012

7.3.3.1.9 SVC_3319 - Additional responses supported by the OFL Service Gate #5

Test ID	SVC_3319
Test objectives	To verify that the SPBL supports the additional response: eSPBL_E_INVALID_SPBM_CERT as defined in Table 7.4 of ETSI TS 103 666-2 [10].
Configuration reference	CSVC_311, CSVC_312
Initial conditions	
Pipe sessions are opened and established as defined in PSVC_321 and PSVC_322. The pre-configured SPB container indicates that the received SPBM certificate (chain) is not valid.	
Test sequence	
Step	**Description**	**Requirements**
1	The SPBL application gate sends the SET_PARAMETER command for storing the SPBM credentials parameters from the Service Provider in the SPBL service gate.	
2	The SPBL service gate returns ANY_OK if the command is successfully executed.	
3	The SPBL application gate sends the GET_SPB_METADATA command.	
4	The SPBL service gate sends the response value '14' (eSPBL_E_INVALID_SPBM_CERT).	RQ0703_013

7.3.3.1.10 SVC_33110 - Additional responses supported by the OFL Service Gate #6

Test ID	SVC_33110
Test objectives	To verify that the SPBL supports the additional response: eSPBL_E_EXCEED_TELECOM_CAPABILITY as defined in Table 7.4 of ETSI TS 103 666-2 [10].
Configuration reference	CSVC_311, CSVC_312
Initial conditions	
Pipe sessions are opened and established as defined in PSVC_321 and PSVC_322. The pre-configured SPB container holds a number of Telecom Secondary Platform Bundles higher than the limit indicated in the TELECOM_CAPABILITY.	

Test sequence		
Step	**Description**	**Requirements**
1	The SPBL application gate sends the SET_PARAMETER command for storing the SPBM credentials parameters from the Service Provider in the SPBL service gate.	
2	The SPBL service gate returns ANY_OK if the command is successfully executed.	
3	The SPBL application gate sends the SWITCH_TELECOM_SPB command.	
4	The SPBL service gate sends the response value '15' eSPBL_E_EXCEED_TELECOM_CAPABILITY.	RQ0703_014

7.3.4 Requirements not testable, implicitly verified or verified elsewhere

7.3.4.1 OFL service

The following requirements, identified in ETSI TS 103 666-2 [10] are not tested in accordance with the present document, as they are referencing requirements from another standardization body (GlobalPlatform):

RQ0703_006, RQ0703_008.

The following requirement, identified in ETSI TS 103 666-2 [10] is not tested in accordance with the present document:

RQ0703_002.

The following requirement, identified in ETSI TS 103 666-2 [10] are tested in the context of clause 9.4 of the present document:

RQ0703_017.

The following requirement, identified in ETSI TS 103 666-2 [10] are tested in the context of clause 12 of the present document:

RQ0703_007, RQ0703_015, RQ0703_016, RQ0703_018.

7.3.4.2 Communication service

The following requirement, identified in ETSI TS 103 666-2 [10] is not tested in accordance with the present document, as it is referencing requirements from another standardization body (GlobalPlatform):

RQ0703_020.

7.3.4.3 Management service

The following requirement, identified in ETSI TS 103 666-2 [10] is not tested in accordance with the present document, as it is referencing requirements from another standardization body (GlobalPlatform):

RQ0703_021, RQ0703_022, RQ0703_023.

7.4 Cryptographic functions

7.4.1 Configurations

There are no specific configurations defined for cryptographic functions testing.

7.4.2 Procedures

There are no specific procedures defined for cryptographic functions testing.

7.4.3 Test descriptions

There are no specific test descriptions defined for testing the provisioning of cryptographic functions.

7.4.4 Requirements verified elsewhere

The following requirement, identified in ETSI TS 103 666-2 [10] is not tested in accordance with the present document, as it is referencing requirements from another standardization body (GlobalPlatform):

RQ0704_001.

7.5 Primary Platform identification

7.5.1 Configurations

There are no specific configurations defined for Primary Platform identification testing.

7.5.2 Procedures

There are no specific procedures defined for Primary Platform identification testing.

7.5.3 Test descriptions

There are no specific test descriptions defined for testing the provisioning of Primary Platform identification.

7.5.4 Requirements verified elsewhere

The following requirement, identified in ETSI TS 103 666-2 [10] are tested in the context of clause 12.6 of the present document:

RQ0705_001.

The following requirement, identified in ETSI TS 103 666-2 [10] is not tested in accordance with the present document, as the upgrade of the SPBL is not possible without support of the SSP manufacturer:

RQ0705_002.

7.6 Provisioning of Primary Platform software

7.6.1 Configurations

There are no specific configurations defined for testing the provisioning of primary platform software.

7.6.2 Procedures

There are no specific procedures defined for testing the provisioning of primary platform software.

7.6.3 Test descriptions

There are no specific test descriptions defined for testing the provisioning of Primary Platform software.

7.6.4 Requirements verified elsewhere

The following requirement, identified in ETSI TS 103 666-2 [10] is not tested in accordance with the present document, as it is referencing requirements from another standardization body (GlobalPlatform):

RQ0706_001.

7.7 Part Number Identifier

7.7.1 Configurations

There are no specific configurations defined for testing the part number identifier.

7.7.2 Procedures

There are no specific procedures defined for testing the part number identifier.

7.7.3 Test descriptions

There are no specific test descriptions defined for testing the Part Number Identifier.

7.7.4 Requirements verified elsewhere

The following requirement, identified in ETSI TS 103 666-2 [10] is implicitly tested with the SSP information tests from clause 12.6.2.2 (aPartNumberId):

RQ0707_001.

8 Primary Platform Interface

8.1 Kernel functions ABI/API

8.1.1 Configurations

There are no specific configurations defined for testing the kernel functions ABI/API.

8.1.2 Procedures

There are no specific procedures defined for testing the kernel functions ABI/API.

8.1.3 Test descriptions

There are no specific test descriptions defined for testing the Kernel functions ABI/API.

8.1.4 Requirements verified elsewhere

The following requirement, identified in ETSI TS 103 666-2 [10] is not tested in accordance with the present document, as it is referencing requirements from another standardization body (GlobalPlatform):

RQ0801_001.

8.2 Communication service interface

8.2.1 Configurations

There are no specific configurations defined for testing the communication service interface.

8.2.2 Procedures

There are no specific procedures defined for testing the communication service interface.

8.2.3 Test descriptions

There are no specific test descriptions defined for testing the Communication service interface.

8.2.4 Requirements verified elsewhere

The following requirement, identified in ETSI TS 103 666-2 [10] is not tested in accordance with the present document, as it is referencing requirements from another standardization body (GlobalPlatform):

RQ0802_001.

8.3 Secondary Platform Bundle management service interface

8.3.1 Configurations

There are no specific configurations defined for testing the Secondary Platform Bundle management service interface.

8.3.2 Procedures

There are no specific procedures defined for testing the Secondary Platform Bundle management service interface.

8.3.3 Test descriptions

There are no specific test descriptions defined for testing the Secondary Platform Bundle management service interface.

8.3.4 Requirements verified elsewhere

The following requirement, identified in ETSI TS 103 666-2 [10] is not tested in accordance with the present document:

RQ0803_001.

9 Secondary Platform Bundle

9.1 Introduction

There are no test requirements identified in the respective clause in ETSI TS 103 666-2 [10].

9.2 States

9.2.1 Configurations

There are no specific configurations defined for testing the Secondary Platform Bundle states.

9.2.2 Procedures

There are no specific procedures defined for testing the Secondary Platform Bundle states.

9.2.3 Test descriptions

There are no specific test descriptions defined for testing the Secondary Platform Bundle states.

9.2.4 Requirements not testable, implicitly verified or verified elsewhere

The following requirements, identified in ETSI TS 103 666-2 [10] are not tested in accordance with the present document, as they are referencing requirements from another standardization body (GlobalPlatform):

RQ0902_001, RQ0902_002, RQ0902_007.

The following requirements, identified in ETSI TS 103 666-2 [10] are not tested in accordance with the present document, as it is descriptive only:

RQ0902_003, RQ0902_004, RQ0902_005.

The following requirement, identified in ETSI TS 103 666-2 [10] is not tested in accordance with the present document as it is not testable:

RQ0902_006.

9.3 Secondary Platform Bundle container format

9.3.1 Configurations

There are no specific configurations defined for testing the SPB container format.

9.3.2 Procedures

There are no specific procedures defined for testing the SPB container format.

9.3.3 Test descriptions

There are no specific test descriptions defined for testing requirements identified in the Secondary Platform Bundle container format clause.

9.3.4 Requirements not testable

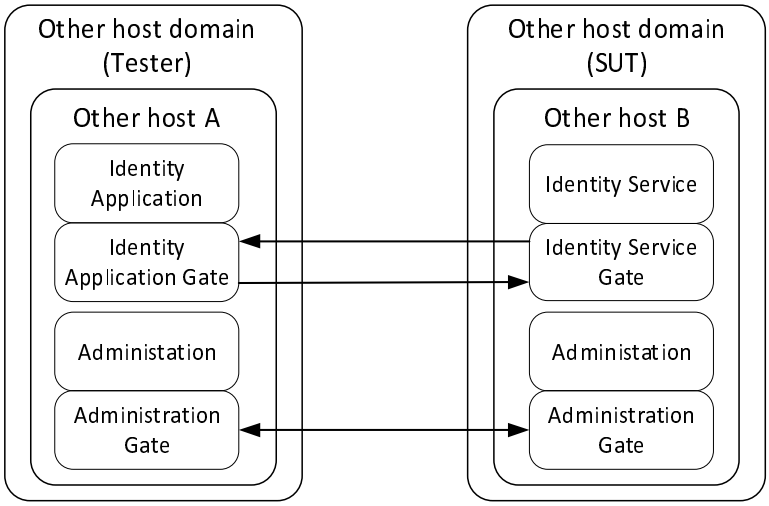
The following requirement, identified in ETSI TS 103 666-2 [10] is not tested in accordance with the present document, as it is referencing requirements from another standardization body (GlobalPlatform):

RQ0903_001.

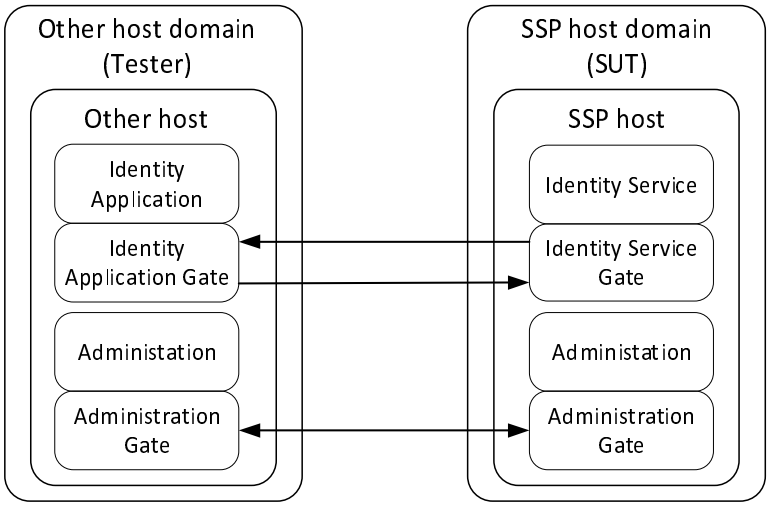
9.4 Secondary Platform

9.4.1 Configurations

9.4.1.1 CiSP_411

Configuration ID	CiSP_411
Configuration description	

9.4.1.2 CiSP_412

Configuration ID	CiSP_412
Configuration description	

9.4.1.3 ASN.1 definition

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,
   SspUiccCapability
   SSPUserInterface
VersionType
FROM SSPDefinitions;
-- ASN1STOP
```

9.4.2 Procedures

9.4.2.1 PiSP_421 – Open a pipe session with the Identity gate of the Terminal host

Procedure ID	PiSP_421
Objectives	To verify that the SSP host is able to open a pipe session to the identity gate of the Terminal host.
Configuration reference	CiSP_411
Initial conditions	
The Terminal host is registered to the SCL network controller host.	
Test sequence	
Step	**Description**
1	The Administration gate in the Other host A (Tester) sends EVT_ADM_BIND to the Administration gate in the Other 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	The Administration gate in the Other host B (SUT) sends EVT_ADM_BIND to the Administration gate in the Other 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).

9.4.2.2 PiSP_422 – Open a pipe session with the Identity gate of the SSP host

Procedure ID	PiSP_422
Objectives	To verify that the Other host is able to open a pipe session to the identity gate of the SSP host.
Configuration reference	CiSP_412
Initial conditions	
The SSP host is registered to the SCL network controller host.	
Test sequence	
Step	**Description**
1	The Administration gate in the Other host (Tester) sends EVT_ADM_BIND to the Administration gate in the SSP 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	The Administration gate in the SSP host (SUT) sends EVT_ADM_BIND to the 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).

9.4.3 Test descriptions

9.4.3.1 High-level OS

The following requirement, identified in ETSI TS 103 666-2 [10] is not tested in accordance with the present document, as it is referencing requirements from another standardization body (GlobalPlatform):

RQ0904_001.

9.4.3.2 Execution framework

The following requirement, identified in ETSI TS 103 666-2 [10] is not tested in accordance with the present document, as it is descriptive only:

RQ0904_002.

9.4.3.3 UICC platform as a Secondary Platform

The following requirement, identified in ETSI TS 103 666-2 [10] refers to UICC APDU functionality defined in ETSI TS 103 666-1 [9]. It shall be verified in accordance with the respective tests defined in ETSI TS 103 999-1 [11] for the APDU protocol (clause 5.6.2 of [11]) and the UICC APDU gate (clause 10.2.8.2 of [11]):

RQ0904_003.

NOTE: Check with the recent version of ETSI TS 103 999-1 [11] if appropriate tests are defined.

9.4.3.4 Capability exchange

9.4.3.4.1 iSP_4341 – Terminal Capabilities (expected)

Test ID	iSP_4341
Test objectives	To verify that, in addition to test descriptions defined in ETSI TS 103 999-1 [11] clause 6.4, data sent by the terminal during the capability exchange procedure is successfully handled by the iSSP if aPhysicalInterfaces data is not included.
Configuration reference	CiSP_411

Initial conditions

The procedure PiSP_421 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	The Identity application gate sends the ANY_GET_PARAMETER command with the register identifier '80' (CAPABILITY_EXCHANGE) to the Identity service gate.	
2	The Identity service gate sends an aResponse to the identity application gate.	RQ0904_004

```
-- 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
```

9.4.3.4.2 iSP_4342 – Terminal Capabilities (unused parameter)

Test ID	iSP_4342
Test objectives	To verify that, in addition to test descriptions defined in ETSI TS 103 999-1 [11] clause 6.4, data sent by the terminal during the capability exchange procedure is successfully handled by the iSSP if aPhysicalInterfaces data is included unexpectedly.
Configuration reference	CiSP_411

Initial conditions

The procedure PiSP_421 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*/
aPHYSICALINTERFACE SEQUENCE ::= 02 / *<STORE(aPHYSICALINTERFACE)> */
/ it indicates which physical interface is available on the terminal */
-- ASN1STOP
```

Test sequence

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

2	The Identity service gate sends an aResponse to the identity application gate.
	`-- ASN1START`
	`aResponse TerminalCapability ::= {`
	`  aTerminalRelease '0000'H,`
	`/*<COMPARE(aTERMINALRELEASE,GT,EQ)>*/`
	`  aTerminalVendorName "0",`
	`/*<ISFIELDNOTEXIST()> OR <COMPARE(aEMPTY_1,DIF)>*/`
	`  aPhysicalInterfaces 02,`
	`/*<COMPARE(aPHYSICALINTERFACE,EQ,GT)>*/`
	`  aInterfacePowerSupply 0,`
	`/*<COMPARE(aINTERFACEPOWERSUPPLY,EQ,GT)>*/`
	`  aExternalPowerSupply 0,`
	`/*<COMPARE(aEXTERNALPOWERSUPPLY,EQ,GT)>*/`
	`  aToolkitTerminalProfile '00'H`
	`/*<ISFIELDNOTEXIST> OR <COMPARE(aEMPTY_2,DIF)>*/`
	`}`
	`-- ASN1STOP`
		RQ0904_005

9.4.3.4.3 iSP_4343 – iSSP Capabilities (expected)

Test ID	iSP_4343
Test objectives	To verify that, in addition to test descriptions defined in ETSI TS 103 999-1 [11] clause 6.4, data sent by the iSSP during the capability exchange procedure includes the expected values as defined in ETSI TS 103 666-2 [10], clause 9.4.4.
Configuration reference	CiSP_412

Initial conditions

The procedure PiSP_422 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	The Identity application gate sends the ANY_GET_PARAMETER command with the register identifier '80' (CAPABILITY_EXCHANGE) to the Identity service gate.	
2	The Identity service gate sends an aResponse to the identity application gate.	RQ0904_006 RQ0904_007

```
-- 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)>*/
  }
}
```

9.4.3.4.4 iSP_4344 – iSSP Capabilities (unused parameter)

Test ID	iSP_4344
Test objectives	To verify that, in addition to test descriptions defined in ETSI TS 103 999-1 [11] clause 6.4, the terminal correctly handles unused parameters sent by the iSSP during the capability exchange procedure.
Configuration reference	CiSP_412

Initial conditions
The procedure PiSP_422 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)> */
aSSPCLASSSPECIFICCAPABILITIES SSPClassSpecificCapabilities ::= eSSPClassSpecificCapabilities /*
<STORE(aSSPCLASSSPECIFICCAPABILITIES)> */
/* it is indicating the specific capabilities (if defined) */
aPHYSICALINTERFACES SSPClassList ::= eSSPClassList /* <STORE(aPHYSICALINTERFACES)> */
/*it is holding the list of supported physical interfaces */
aSSPEXTERNALMAXPOWERCONSUMPTION SSPMaxPower ::= eSSPMaxPower /*
<STORE(aSSPEXTERNALMAXPOWERCONSUMPTION)> */
/* it is indication the maximum power consumption as an integer (0 .. 1000)
*/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	The Identity application gate sends the ANY_GET_PARAMETER command with the register identifier '80' (CAPABILITY_EXCHANGE) to the Identity service gate.	
2	The Identity service gate sends an aResponse to the identity application gate.	RQ0904_008

```
-- ASN1START
aResponse TerminalCapability ::= {
  aTerminalRelease '0000'H,
/*<COMPARE(aTERMINALRELEASE,GT,EQ)>*/
  aTerminalVendorName "0",
/*<ISFIELDNOTEXIST()> OR <COMPARE(aEMPTY_1,DIF)>*/
  aClassSpecificCapabilities 0,
/*<COMPARE(aSSPCLASSSPECIFICCAPABILITIES,EQ,GT)>*/
  aPhysicalInterfaces 03,
/*<COMPARE(aPHYSICALINTERFACES,EQ,GT)>*/
  aSspExternalMaxPowerConsumption 60,
/*<COMPARE(aSSPEXTERNALMAXPOWERCONSUMPTION,EQ,GT)>*/
  aInterfacePowerSupply 0,
/*<COMPARE(aINTERFACEPOWERSUPPLY,EQ,GT)>*/
  aExternalPowerSupply 0,
/*<COMPARE(aEXTERNALPOWERSUPPLY,EQ,GT)>*/
  aToolkitTerminalProfile '00'H
/*<ISFIELDNOTEXIST> OR <COMPARE(aEMPTY_2,DIF)>*/
}
-- ASN1STOP
```

9.4.3.5 Identifiers of Secondary Platform Bundle

The following requirements, identified in ETSI TS 103 666-2 [10] are implicitly verified in installation procedure tests of clause 12.3 of the present document. Public and private identifiers (UUIDs) are included in the Si3.LoadBoundSpbInfo response:

RQ0904_009. RQ0904_010.

9.4.3.6 ASN.1 Stop

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

9.5 SSP Application

9.5.1 Configurations

There are no specific configurations required for testing the SSP Application.

9.5.2 Procedures

There are no specific procedures required for testing the SSP Application.

9.5.3 Test descriptions

There are no specific test descriptions defined for testing the SSP Application.

9.5.4 Requirements not testable

The following requirements, identified in ETSI TS 103 666-2 [10] are not tested in accordance with the present document, as they are descriptive only:

RQ0905_001, RQ0905_002.

9.6 Lifecycle management of Secondary Platform Bundles

9.6.1 Configurations

There are no specific configurations required for testing the lifecycle management of SPBs.

9.6.2 Procedures

There are no specific procedures required for testing the lifecycle management of SPBs.

9.6.3 Test descriptions

There are no specific test descriptions defined for testing the lifecycle management of SPBs.

9.6.4 Requirements implicitly verified or verified elsewhere

The following requirement, identified in ETSI TS 103 666-2 [10] is implicitly verified in the enabling procedure tests for a Secondary Platform Bundle of clause 12.4.1. The decision about enabling the Telecom SPB is made in step 4 of the procedure description in [10]:

RQ0906_001.

9.7 Secondary Platform Bundle family identifier

9.7.1 Configurations

There are no specific configurations required for the SPB family identifier testing.

9.7.2 Procedures

There are no specific procedures required for the SPB family identifier testing.

9.7.3 Test descriptions

There are no specific test descriptions defined for testing requirements identified in the Secondary Platform Bundle family identifier clause.

9.7.4 Requirements not testable

The following requirement, identified in ETSI TS 103 666-2 [10] is not tested in accordance with the present document, as it is descriptive only:

RQ0907_001.

10 Communication interface related testing

10.1 Configurations

For specific configurations required, see ETSI TS 103 999-1 [11] clause 8.

10.2 Procedures

For specific procedure required, see ETSI TS 103 999-1 [11] clause 8.

10.3 Test descriptions

There are no specific test descriptions defined for testing requirements identified in the communication interface clause.

10.4 Requirements verified elsewhere

The following requirement, identified in ETSI TS 103 666-2 [10] is not tested in accordance with the present document:

RQ1002_002.

The requirements RQ1002_001 and RQ1002_003, identified in ETSI TS 103 666-2 [10] are implicitly tested when tests in accordance with ETSI TS 103 999-1 [11] clause 8 are executed.

11 Certification related testing

11.1 Configurations

There are no specific configurations required for certification related testing.

11.2 Procedures

There are no specific procedures required for certification related testing.

11.3 Test descriptions

There are no specific test descriptions defined for testing requirements identified in the certification clause.

11.4 Requirements verified elsewhere

11.4.1 Introduction

The following requirement, identified in ETSI TS 103 666-2 [10] is not tested in accordance with the present document:

RQ1101_001.

11.4.2 Primary Platform certification

The following requirements, identified in ETSI TS 103 666-2 [10] are not tested in accordance with the present document:

RQ1102_001, RQ1102_002.

12 iSSP ecosystem and interfaces related testing

12.1 General architecture

There are no requirements for the general architecture identified in ETSI TS 103 666-2 [10].

12.2 Security overview

12.2.1 Public key infrastructure for Si4 interface

12.2.1.1 Configurations

The configurations defined for Si4 interface testing as defined in clause 12.6.6.1 are used for public key infrastructure for Si4 interface testing.

12.2.1.2 Procedures

The procedures defined for Si4 interface testing as defined in clause 12.6.6.2 are used for public key infrastructure for Si4 interface testing.

12.2.1.3 Test descriptions

There are no specific test descriptions defined for public key infrastructure for Si4 interface testing.

12.2.1.4 Requirements not testable, implicitly verified or verified elsewhere

The following requirements are tested with the Si4 interface functions. The requirements will show up in the 'Requirement' column of the test step the fulfillment of the requirement can be verified with:

RQ1202_001, RQ1202_002, RQ1202_003, RQ1202_004, RQ1202_006, RQ1202_007, RQ1202_008, RQ1202_009, RQ1202_010, RQ1202_011, RQ1202_012, RQ1202_013, RQ1202_014, RQ1202_015, RQ1202_016, RQ1202_017, RQ1202_018, RQ1202_019, RQ1202_020, RQ1202_021, RQ1202_022, RQ1202_023, RQ1202_024, RQ1202_025, RQ1202_026, RQ1202_027, RQ1202_028, RQ1202_030, RQ1202_033.

The requirements for the public key infrastructure for the Si4 interface identified in ETSI TS 103 666-2 [10] are descriptive. The following requirements, identified in ETSI TS 103 666-2 [10] are not tested in accordance with the present document, as their implicit verification by executing Si4 or Si2 test cannot be guaranteed:

RQ1202_005, RQ1202_029, RQ1202_031, RQ1202_032, RQ1202_034, RQ1202_035, RQ1202_036, RQ1202_037.

12.2.2 Cryptographic algorithms

12.2.2.1 Configurations

The configurations defined for Si4 interface testing as defined in clause 12.6.6.1 are used for testing cryptographic algorithms on the Si4 interface.

12.2.2.2 Procedures

The procedures defined for Si4 interface testing as defined in clause 12.6.6.2 are used for testing cryptographic algorithms on the Si4 interface.

12.2.2.3 Test descriptions

There are no specific test descriptions defined for testing cryptographic algorithms public with the Si4 interface.

12.2.2.4 Requirements not testable, implicitly verified or verified elsewhere

The following requirements are tested with the Si4 interface functions. The requirements will show up in the 'Requirement' column of the test step the fulfillment of the requirement can be verified with:

RQ1202_038, RQ1202_039, RQ1202_040, RQ1202_041, RQ1202_042.

12.3 Secondary Platform Bundle provisioning procedure

12.3.1 Overview

12.3.1.1 Configuration

There are no specific configurations required defined for testing requirements defined in the overview clause.

12.3.1.2 Procedures

There are no specific procedures defined for testing requirements defined in the overview clause.

12.3.1.3 Test descriptions

There are no specific test descriptions defined to procedures for testing requirements identified in the overview clause.

12.3.1.4 Requirements not testable, implicitly verified or verified elsewhere

Clause 12.3.1 of ETSI TS 103 666-2 [10] is the 'Overview' clause for the description of the Secondary Platform Bundle provisioning procedure.

The following requirements are descriptive and can be seen as implicitly verified if the Si1, Si2 and Si3 related tests as defined in the respective subclauses of clause 12.6 of the present document can be executed successfully:

RQ1203_001, RQ1203_002, RQ1203_003.

The following requirement are descriptive and related to the preparation of the Secondary Platform Bundle. As the preparation of the Secondary Platform Bundle cannot be done by the tester the following requirements cannot be verified and are out of scope of the present document:

RQ1203_004, RQ1203_005.

12.3.2 Preparation procedure

12.3.2.1 Configuration

There are no specific configurations defined for testing the preparation procedure.

12.3.2.2 Procedures

There are no specific procedures defined for testing the preparation procedure.

12.3.2.3 Test descriptions

There are no specific test descriptions defined for testing requirements identified in the preparation procedure clause.

12.3.2.4 Requirements not testable, implicitly verified or verified elsewhere

Clause 12.3.2 of ETSI TS 103 666-2 [10] describes the preparation procedure to select a Secondary Platform Bundle.

The requirements in this clause are descriptive.

The following requirements can be seen as implicitly verified if the Secondary Platform Bundle selected for testing is matching the terminal and the SSP capabilities:

RQ1203_006, RQ1203_007.

The following requirement is tested in the Si1.SelectSpb related test descriptions. Requirements will show up in the ′Requirement′ column of the test step the fulfillment of the requirement can be verified with:

RQ1203_008.

The following requirement is tested in the Si1.SelectSpb and Si1.FinalizePreparation related test descriptions. The requirements will show up in the ′Requirement′ column of the test step the fulfillment of the requirement can be verified with:

RQ1203_009.

The following requirements are tested in the Si1.CreateSPReference related test descriptions. Requirements will show up in the ′Requirement′ column of the test step the fulfillment of the requirement can be verified with:

RQ1203_010, RQ1203_011a, RQ1203_011b.

The following requirement is tested in the Si1.CancelPreparation related test description where a CodeM is used. The requirements will show up in the ′Requirement′ column of the test step the fulfillment of the requirement can be verified with:

RQ1203_012.

12.3.3 Download procedure

12.3.3.1 Configuration

There are no specific configurations defined for testing the download procedure.

12.3.3.2 Procedures

There are no specific procedures defined for testing the download procedure.

12.3.3.3 Test descriptions

There are no specific test descriptions defined for testing requirements identified in the download procedure clause.

12.3.3.4 Requirements not testable, implicitly verified or verified elsewhere

Clause 12.3.3 of ETSI TS 103 666-2 [10] describes the download procedure for a Secondary Platform Bundle.

The following requirements are out of scope of the present document, as they should be executed using keys and certificates not dedicated to testing services:

RQ1203_013, RQ1203_014, RQ1203_015, RQ1203_016, RQ1203_021, RQ1203_024.

The following requirements are out of scope, as LBA application functionality cannot be tested in accordance to the present document:

RQ1203_023, RQ1203_025, RQ1203_026, RQ1203_027.

The functionality the following requirements are based on can be seen as implicitly verified if the related Si3 tests can be executed successfully:

- RQ1203_017, RQ1203_018, RQ1203_019, RQ1203_020, RQ1203_022 with Si3.GetSspInfo testing,
- RQ1203_028, RQ1203_029, RQ1203_030 with Si3.SetSpbmCredential testing,
- RQ1203_031, RQ1203_032 with Si3.GetSspCredential testing.

The functionality the following requirements are based on can be seen as implicitly verified if the related Si2.GetBoundSpbImage tests can be executed successfully:

RQ1203_033, RQ1203_034, RQ1203_035, RQ1203_036.

NOTE: All tests indicating that 'functionality' is tested, imply that the usage of 'live' keys and certificates may lead to deviations.

12.3.4 Installation procedure

12.3.4.1 Configuration

There are no specific configurations defined for testing the installation procedure.

12.3.4.2 Procedures

There are no specific procedures defined for testing the installation procedure.

12.3.4.3 Test descriptions

There are no specific test descriptions defined for testing requirements identified in the installation procedure clause.

12.3.4.4 Requirements not testable, implicitly verified or verified elsewhere

The functionality the following requirements are based on can be seen as implicitly verified if the related Si3.LoadBoundSpbInfo tests can be executed successfully:

RQ1203_037, RQ1203_038, RQ1203_039, RQ1203_040.

NOTE: All tests indicating that 'functionality' is tested, imply that the usage of 'live' keys and certificates may lead to deviations.

12.3.5 SSP activation code

12.3.5.1 Configuration

There are no specific configurations defined for testing the SSP activation code.

12.3.5.2 Procedures

There are no specific procedures defined for testing the SSP activation code.

12.3.5.3 Test descriptions

There are no specific test descriptions defined for testing requirements identified in the SSP activation code clause.

12.3.5.4 Requirements not testable, implicitly verified or verified elsewhere

The following requirements are out of scope, as LBA application functionality cannot be tested in accordance to the present document:

RQ1203_041, RQ1203_042.

12.4 Secondary Platform Bundle management procedure

12.4.1 Enable a Secondary Platform Bundle

12.4.1.1 Configuration

There are no specific configurations defined for testing requirements defined in the enable a Secondary Platform Bundle clause.

12.4.1.2 Procedures

There are no specific procedures defined for testing requirements defined in the enable a Secondary Platform Bundle clause.

12.4.1.3 Test descriptions

There are no specific test descriptions defined for testing requirements identified in the SSP activation code clause.

12.4.1.4 Requirements not testable, implicitly verified or verified elsewhere

Clause 12.4.1 of ETSI TS 103 666-2 [10] describes the enabling of a Secondary Platform Bundle.

The following requirements are related to user action or ′user intent′ and therefore out of scope of the present document:

RQ1204_001, RQ1204_002.RQ1204_003.

The following requirements are tested with the Si3 interface functions. Requirements will show up in the ′Requirement′ column of the test step the fulfillment of the requirement can be verified with:

RQ1204_004, RQ1204_005.

12.4.2 Disable a Secondary Platform Bundle

12.4.2.1 Configuration

There are no specific configurations defined for testing requirements defined in the disable a Secondary Platform Bundle clause.

12.4.2.2 Procedures

There are no specific procedures defined for testing requirements defined in the disable a Secondary Platform Bundle clause.

12.4.2.3 Test descriptions

There are no specific test descriptions defined for testing requirements identified in the disable a Secondary Platform Bundle clause.

12.4.2.4 Requirements not testable, implicitly verified or verified elsewhere

Clause 12.4.2 of ETSI TS 103 666-2 [10] describes the disabling of a Secondary Platform Bundle.

The following requirements are related to user action or ′user intent′ and therefore out of scope of the present document:

RQ1204_006, RQ1204_007, RQ1204_008.

The following requirement is tested with the Si3 interface functions. The requirement will show up in the ′Requirement′ column of the test step the fulfillment of the requirement can be verified with:

RQ1204_009.

12.4.3 Delete a Secondary Platform Bundle

12.4.3.1 Configuration

There are no specific configurations defined for testing requirements defined in the delete a Secondary Platform Bundle clause.

12.4.3.2 Procedures

There are no specific procedures defined for testing requirements defined in the delete a Secondary Platform Bundle clause.

12.4.3.3 Test descriptions

There are no specific test descriptions defined for testing requirements identified in the delete a Secondary Platform Bundle clause.

12.4.3.4 Requirements not testable, implicitly verified or verified elsewhere

Clause 12.4.3 of ETSI TS 103 666-2 [10] describes the deletion of a Secondary Platform Bundle.

The following requirements are related to user action or ′user intent′ and therefore out of scope of the present document:

RQ1204_010, RQ1204_011, RQ1204_012.

The following requirements are tested with the Si3 interface functions. The requirements will show up in the ′Requirement′ column of the test step the fulfillment of the requirement can be verified with:

RQ1204_013, RQ1204_014, RQ1204_015.

12.4.4 SPB metadata retrieving procedure

12.4.4.1 Configuration

There are no specific configurations defined for testing requirements defined in the SPB metadata retrieving procedure clause.

12.4.4.2 Procedures

There are no specific procedures defined for testing requirements defined in the SPB metadata retrieving procedure clause.

12.4.4.3 Test descriptions

There are no specific test descriptions defined for testing requirements identified in the SPB metadata retrieving procedure clause.

12.4.4.4 Requirements not testable, implicitly verified or verified elsewhere

Clause 12.4.4 of ETSI TS 103 666-2 [10] describes the procedure used to retrieve SPB metadata.

The following requirement is related to user action or ′user intent′ and therefore out of scope of the present document:

RQ1204_016.

The following requirements are tested with the Si3 interface functions. The requirements will show up in the ′Requirement′ column of the test step the fulfillment of the requirement can be verified with:

RQ1204_017, RQ1204_018.

12.4.5 SPB state retrieving procedure

12.4.5.1 Configuration

There are no specific configurations defined for testing requirements defined in the SPB state retrieving procedure clause.

12.4.5.2 Procedures

There are no specific procedures defined for testing requirements defined in the SPB state retrieving procedure clause.

12.4.5.3 Test descriptions

There are no specific test descriptions defined for testing requirements identified in the SPB state retrieving procedure clause.

12.4.5.4 Requirements not testable, implicitly verified or verified elsewhere

Clause 12.4.5 of ETSI TS 103 666-2 [10] describes the procedure used to retrieve the SPB state.

The following requirements are tested with the Si3 interface functions. The requirements will show up in the ′Requirement′ column of the test step the fulfillment of the requirement can be verified with:

RQ1204_019, RQ1204_022, RQ1204_023.

The following requirements are related to preparation steps and can therefore not be verified explicitly:

RQ1204_020, RQ1204_021.

These requirements can be seen as verified when test SI3_653121 - Si3.GetSpbState is executed successfully.

NOTE: A successful execution of test SI3_653121 - Si3.GetSpbState implies the successful execution of test SI3_653111 – Si3.GetSspInfo.

12.5 Notification procedure

12.5.1 Overview

There are no requirements for this overview clause identified in ETSI TS 103 666-2 [10].

12.5.2 Notification of the service provider

12.5.2.1 Configurations

There are no specific configurations defined for testing the notification of the service provider.

12.5.2.2 Procedures

There are no specific procedures defined for testing the notification of the service provider.

12.5.2.3 Test descriptions

There are no specific test descriptions defined for testing requirements identified in the notification procedure clause.

12.5.2.4 Requirements not testable, implicitly verified or verified elsewhere

Clause 12.5.2 of ETSI TS 103 666-2 [10] describes the functions used to notify of the service provider.

The following requirements can be seen as implicitly verified if the Si1.HandleNotification testing has been executed successfully:

RQ1205_001, RQ1205_002.

12.5.3 Notification from the LBA

12.5.3.1 Configurations

There are no specific configurations defined for testing the notification from the LBA.

12.5.3.2 Procedures

There are no specific procedures defined for testing the notification from the LBA.

12.5.3.3 Test descriptions

There are no specific test descriptions defined for testing requirements identified in the notification from the LBA clause.

12.5.3.4 Requirements not testable, implicitly verified or verified elsewhere

Clause 12.5.3 of ETSI TS 103 666-2 [10] describes the functions used for notifications from the LBA.

Testing an application like the Local Bundle Assistant is out of scope of the present document. The following requirements cannot be verified:

RQ1205_003, RQ1205_004, RQ1205_004, RQ1205_005, RQ1205_006, RQ1205_007, RQ1205_008, RQ1205_009 and RQ1205_010.

12.6 Interfaces and functions

12.6.1 Overview

12.6.1.1 Configurations

There are no specific configurations defined for testing requirements defined in the overview clause.

12.6.1.2 Procedures

There are no specific procedures defined for testing requirements defined in the overview clause.

12.6.1.3 Test descriptions

There are no specific test descriptions defined for testing requirements identified in the overview clause.

12.6.1.4 Requirements not testable, implicitly verified or verified elsewhere

Clause 12.6.1 of ETSI TS 103 666-2 [10] describes the interfaces and the functions used for the Secondary Platform Bundle provisioning and the Secondary Platform Bundle management operations.

The requirements RQ1206_001, RQ1206_002, RQ1206_003, RQ1206_004 identified in ETSI TS 103 666-2 [10] are unspecific as they list functions of the different interfaces only. Thus, the requirements are not explicitly mentioned, but tested in detail with the interface related test described in the present document:

- RQ1206_001 and RQ1206_002 with Si1 testing,
- RQ1206_003 with Si2 testing,
- RQ1206_004 with Si3 testing.

12.6.2 Common features

12.6.2.1 Configurations

There are no specific configurations defined for testing requirements defined in the common features of interfaces and functions.

12.6.2.2 Procedures

There are no specific procedures defined for testing requirements defined in the common features of interfaces and functions.

12.6.2.3 Test descriptions

There are no specific test descriptions defined for testing requirements identified in the common features of interfaces and functions clause.

12.6.2.4 Requirements not testable, implicitly verified or verified elsewhere

12.6.2.4.1 Common data types

The common ASN.1 types and objects defined in clause 12.6.2.1 of ETSI TS 103 666-2 [10] are reflected in the ASN.1 coding used for verification of ASN.1 coded test descriptions of the present document.

12.6.2.4.2 SSP information

The requirements for this clause as identified in ETSI TS 103 666-2 [10] are descriptive and implicitly covered by testing the Si4 interface. The Si4 interface is a composite interface. Si4 interface testing is based on the Si2 and Si3 test descriptions.

The following requirements, identified in ETSI TS 103 666-2 [10] are implicitly tested by executing to the test descriptions defined in clause 12.6.6:

RQ1206_005, RQ1206_006, RQ1206_007, RQ1206_008, RQ1206_009, RQ1206_010, RQ1206_011, RQ1206_012, RQ1206_013, RQ1206_014, RQ1206_015, RQ1206_016, RQ1206_017, RQ1206_018, RQ1206_019, RQ1206_020.

12.6.2.4.3 SPBM credential

The requirements for SPBM credentials as identified in ETSI TS 103 666-2 [10] are descriptive and implicitly covered by testing the Si4 interface.

The following requirements, identified in ETSI TS 103 666-2 [10] are implicitly tested by executing to the test descriptions defined in clause 12.6.6:

RQ1206_021, RQ1206_022, RQ1206_023, RQ1206_024, RQ1206_025, RQ1206_026, RQ1206_027, RQ1206_028, RQ1206_029.

12.6.2.4.4 SSP credential

The requirements for SSP credentials as identified in ETSI TS 103 666-2 [10] are descriptive and implicitly covered by testing the Si4 interface.

The following requirements, identified in ETSI TS 103 666-2 [10] are implicitly tested by executing to the test descriptions defined in clause 12.6.6:

RQ1206_030, RQ1206_031, RQ1206_032, RQ1206_033, RQ1206_034, RQ1206_035, RQ1206_036, RQ1206_037.

12.6.2.4.5 Bound SPB image

The requirements for this bound SPB image as identified in ETSI TS 103 666-2 [10] are descriptive and implicitly covered by testing the Si4 interface.

The following requirements, identified in ETSI TS 103 666-2 [10] are implicitly tested by executing to the test descriptions defined in clause 12.6.6:

RQ1206_038, RQ1206_039, RQ1206_040, RQ1206_041, RQ1206_042, RQ1206_043, RQ1206_044, RQ1206_045, RQ1206_046, RQ1206_047, RQ1206_048, RQ1206_049, RQ1206_050, RQ1206_051, RQ1206_052.

12.6.2.4.6 SPB metadata

The requirements for SPB metadata as identified in ETSI TS 103 666-2 [10] are descriptive and implicitly covered by testing the Si4 interface.

The following requirements, identified in ETSI TS 103 666-2 [10] are implicitly tested by executing to the test descriptions defined in clause 12.6.6:

RQ1206_053, RQ1206_054, RQ1206_055, RQ1206_056.

12.6.2.4.7 Terminal information

The requirements for terminal information as identified in ETSI TS 103 666-2 [10] are descriptive and implicitly covered by testing the Si4 interface.

The following requirements, identified in ETSI TS 103 666-2 [10] are implicitly tested by executing to the test descriptions defined in clause 12.6.6:

RQ1206_057, RQ1206_058, RQ1206_059, RQ1206_060, RQ1206_061.

12.6.2.4.8 Notification token

The requirements for the notification token as identified in ETSI TS 103 666-2 [10] are descriptive and covered by testing the Si4 interface.

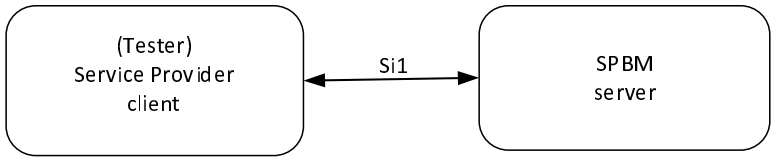
The following requirements are tested with the Si4 interface functions as defined in clause 12.6.6. The requirements will show up in the ′Requirement′ column of the test step the fulfillment of the requirement can be verified with:

RQ1206_062, RQ1206_063, RQ1206_064, RQ1206_065, RQ1206_066, RQ1206_067, RQ1206_068, RQ1206_069, RQ1206_070, RQ1206_071.

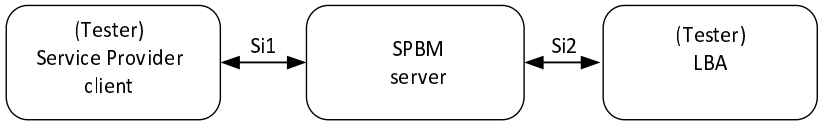
12.6.3 Si1 interface

12.6.3.1 Configurations

12.6.3.1.1 CSI1_6311 – Service Provider - SPB Manager

Configuration ID	CSI1_6311
Configuration description	

12.6.3.1.2 CSI1_6312 – Service Provider - SPB Manager - SPBL

Configuration ID	CSI1_6312
Configuration description	

12.6.3.1.3 ASN1 definition

```
-- ASN1START
SSPSI1configurations { itu-t (0) identified-organization (4) etsi (0) smart-secure-platform (3666)
part2 (2) test (2) si1 ( 1) }
DEFINITIONS
AUTOMATIC TAGS
EXTENSIBILITY IMPLIED ::=
BEGIN
EXPORTS ALL;
/* Imports */
IMPORTS
   UUID,
   MetaData,
   CodeM,
   Si1CancelPreparationResponse,
   Si1CancelPreparationCommand,
   Si1FinalizePreparationResponse,
   Si1FinalizePreparationCommand,
   Si1SelectSpbResponse,
   Si1SelectSpbCommand,
   Si1CreateSPReferenceResponse,
   Si1CreateSPReferenceCommand,
   Si1HandleNotificationBlock
FROM ISSPDefinitions;

eFUNCTION-REQUESTER-ID-1     OCTET STRING::='AAAAAA'H
eFUNCTION-REQUESTER-ID-2     OCTET STRING::='BBBBBB'H

eFUNCTION-CALL-ID-SELECT-1   OCTET STRING::='11111111'H
eFUNCTION-CALL-ID-SELECT-2   OCTET STRING::='11111112'H
eFUNCTION-CALL-ID-SELECT-3   OCTET STRING::='11111113'H
eFUNCTION-CALL-ID-SELECT-4   OCTET STRING::='11111114'H
eFUNCTION-CALL-ID-SELECT-5   OCTET STRING::='11111115'H
eFUNCTION-CALL-ID-SELECT-6   OCTET STRING::='11111116'H
eFUNCTION-CALL-ID-SELECT-7   OCTET STRING::='11111117'H
eFUNCTION-CALL-ID-SELECT-8   OCTET STRING::='11111118'H
eFUNCTION-CALL-ID-SELECT-9   OCTET STRING::='11111119'H
```

```
eFUNCTION-CALL-ID-SELECT-10  OCTET STRING::='11111120'H
eFUNCTION-CALL-ID-SELECT-11  OCTET STRING::='11111121'H
eFUNCTION-CALL-ID-SELECT-12  OCTET STRING::='11111122'H
eFUNCTION-CALL-ID-SELECT-13  OCTET STRING::='11111123'H
eFUNCTION-CALL-ID-SELECT-14  OCTET STRING::='11111124'H
eFUNCTION-CALL-ID-SELECT-15  OCTET STRING::='11111125'H
eFUNCTION-CALL-ID-SELECT-16  OCTET STRING::='11111126'H
eFUNCTION-CALL-ID-SELECT-17  OCTET STRING::='11111127'H
eFUNCTION-CALL-ID-SELECT-18  OCTET STRING::='11111128'H
eFUNCTION-CALL-ID-SELECT-19  OCTET STRING::='11111129'H

eFUNCTION-CALL-ID-CREATEREFERENCE-1   OCTET STRING::='11121111'H
eFUNCTION-CALL-ID-CREATEREFERENCE-2   OCTET STRING::='11121112'H
eFUNCTION-CALL-ID-CREATEREFERENCE-3   OCTET STRING::='11121113'H
eFUNCTION-CALL-ID-CREATEREFERENCE-4   OCTET STRING::='11121114'H
eFUNCTION-CALL-ID-CREATEREFERENCE-5   OCTET STRING::='11121115'H
eFUNCTION-CALL-ID-CREATEREFERENCE-6   OCTET STRING::='11121116'H
eFUNCTION-CALL-ID-CREATEREFERENCE-7   OCTET STRING::='11121117'H
eFUNCTION-CALL-ID-CREATEREFERENCE-8   OCTET STRING::='11121118'H
eFUNCTION-CALL-ID-CREATEREFERENCE-9   OCTET STRING::='11121119'H
eFUNCTION-CALL-ID-CREATEREFERENCE-10  OCTET STRING::='11121120'H

eFUNCTION-CALL-ID-FINALIZEPREP-1   OCTET STRING::='11131111'H
eFUNCTION-CALL-ID-FINALIZEPREP-2   OCTET STRING::='11131112'H
eFUNCTION-CALL-ID-FINALIZEPREP-3   OCTET STRING::='11131113'H
eFUNCTION-CALL-ID-FINALIZEPREP-4   OCTET STRING::='11131114'H
eFUNCTION-CALL-ID-FINALIZEPREP-5   OCTET STRING::='11131115'H
eFUNCTION-CALL-ID-FINALIZEPREP-6   OCTET STRING::='11131116'H

eFUNCTION-CALL-ID-CANCELPREP-1     OCTET STRING::='11141111'H
eFUNCTION-CALL-ID-CANCELPREP-2     OCTET STRING::='11141112'H
eFUNCTION-CALL-ID-CANCELPREP-3     OCTET STRING::='11141113'H
eFUNCTION-CALL-ID-CANCELPREP-4     OCTET STRING::='11141114'H
eFUNCTION-CALL-ID-CANCELPREP-5     OCTET STRING::='11141115'H
eFUNCTION-CALL-ID-CANCELPREP-6     OCTET STRING::='11141116'H
eFUNCTION-CALL-ID-CANCELPREP-7     OCTET STRING::='11141117'H
eFUNCTION-CALL-ID-CANCELPREP-8     OCTET STRING::='11141118'H
eFUNCTION-CALL-ID-CANCELPREP-9     OCTET STRING::='11141119'H

eSPBID-1       UUID::='42D07EAFE3C0499DA29C080E63DE8245'H -- this SPB is available in SPBM
eSPBID-2       UUID::='4A4D006277624D62916A7E5802D3D8F6'H -- this SPB is available in SPBM
eSPBID-3       UUID::='808AFE32B21343ED9FEC76B62EA1A52C'H -- this SPB is available in SPBM
eSPBID-4       UUID::='4B54C81F3FF74716B7583C87AA42EE42'H -- this SPB is available in SPBM
eSPBID-5       UUID::='25D2031828F34A5C9EA92D967BC33331'H -- this SPB is available in SPBM
eSPBID-6       UUID::='7B79EDA06419427E8A7AC96503983795'H -- this SPB is available in SPBM
eSPBID-7       UUID::='8727D84A5434413CAC67844EE6CFFB6C'H -- this SPB is available in SPBM
eSPBID-8       UUID::='462ACC6F62744ADB8B4237B37D256096'H -- this SPB is available in SPBM
eSPBID-9       UUID::='9EA66FAC04A841599A93DE337694A120'H -- this SPB is available in SPBM
eSPBID-10      UUID::='DB947AE9189A4165910616C32076BF6C'H -- this SPB is available in SPBM

eSPBID-UNKNOWN    UUID::='31DD86C7A5134E35BCBF83BF92094E9B'H -- this SPB does not exist in SPBM

ePPIDENTIFIER-1      OCTET STRING::='AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA'H

eSPBTYPE-1     OCTET STRING::='111111'H
eSPBTYPE-2     OCTET STRING::='222222'H
eSPBTYPE-7     OCTET STRING::='777777'H
eSPBTYPE-8     OCTET STRING::='888888'H
eSPBTYPE-9     OCTET STRING::='999999'H
eSPBTYPE-10    OCTET STRING::='AAAAAA'H

eSPBTYPE-UNKNOWN     OCTET STRING::='FFFFFF'H

eCODEM-1          CodeM::='00000000111111112222222233333331'H
eCODEM-2          CodeM::='00000000111111112222222233333332'H
eCODEM-3          CodeM::='00000000111111112222222233333333'H
eCODEM-4          CodeM::='00000000111111112222222233333334'H
eCODEM-5          CodeM::='00000000111111112222222233333335'H
eCODEM-6          CodeM::='00000000111111112222222233333336'H
eCODEM-7          CodeM::='00000000111111112222222233333337'H
eCODEM-8          CodeM::='00000000111111112222222233333338'H
eCODEM-9          CodeM::='00000000111111112222222233333339'H
eCODEM-10         CodeM::='0000000011111111222222223333333A'H

eCODEM-UNLINKED   CodeM::='DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD'H
eCODEM-NOTKNOWN   CodeM::='EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE'H
eCODEM-UNKNOWN    CodeM::='FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF'H
```

```
-- ASN1STOP
```

12.6.3.1.4 SPBM configuration

The SPBM under test shall be configured by the SPBM vendor with the following data:

eSPBID-1 configured as eSPBTYPE-1

eSPBID-2 configured as eSPBTYPE-2

eSPBID-3 (no sbpType is configured)

eSPBID-4 (no sbpType is configured)

eSPBID-5 (no sbpType is configured)

eSPBID-6 (no sbpType is configured)

eSPBID-7 configured as eSPBTYPE-7

eSPBID-8 configured as eSPBTYPE-8

eSPBID-9 configured as eSPBTYPE-9

eSPBID-10 configured as eSPBTYPE-10

eCODEM-3 (unlinked)

eCODEM-UNLINKED (unlinked)

no task type is allowed for eFUNCTION-REQUESTER-ID-2

eSi1TaskType-DownloadSPB is allowed for eFUNCTION-REQUESTER-ID-1

12.6.3.2 Procedures

12.6.3.2.1 PSI1_6321 - Open pipe session between service provider and SBP Manager

Procedure ID	PSI1_6321
Objectives	The service provider shall be able to open a session by using the Si1 interface to the SPBM as defined in the clause 12.6.3 of [10].
Configuration reference	CSI1_6311
Initial conditions	
Test sequence	
Step	**Description**
1	The service provider client shall be in charge of managing the connection establishment to the SPBM server for the Si1 interface. The binding of the Si1 interface shall be performed over Hypertext Transfer Protocol version 2 (HTTP/2) as defined in IETF RFC 7540 [26] and the Transport Layer Security (TLS) version 1.3 or higher in mutual authentication mode as defined in IETF RFC 8446 [27].

12.6.3.2.2 PSI1_6322 - Open pipe session between LBA and SBP Manager

Procedure ID	PSI1_6322
Objectives	The LBA shall be in charge of managing the connection establishment to the SPBM server for the Si2 interface. The binding of the Si2 interface shall be performed over Hypertext Transfer Protocol version 2 (HTTP/2) as defined in IETF RFC 7540 [26] and the Transport Layer Security (TLS) version 1.3 or higher in mutual authentication mode as defined in IETF RFC 8446 [27].
Configuration reference	CSI1_6312

Initial conditions	
Test sequence	
Step	**Description**
1	The LBA establishes the Si2 session.

12.6.3.3 Test descriptions

12.6.3.3.1 Si1.CreateSPReference command and response handling

12.6.3.3.1.1 SI1_63311 - Si1.CreateSPReference succeed

Test ID	SI1_63311	
Test objectives	To verify that the service provider client is able to create a reference (CodeM) shared between the service provider client and the SPBM server for a specific SPB ID.	
Configuration reference	CSI1_6311	
Initial conditions		
The procedure PSI1_6321 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**

1	The service provider client sends aSI1-63311-command-01 to the SPBM server:
	`-- ASN1START`
	`aSI1-63311-command-01 Si1CreateSPReferenceCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-CREATEREFERENCE-1`
	`  },`
	`  aCodeM eCODEM-1,`
	`  aSpbId eSPBID-1,`
	`  aTaskType eSi1TaskType-DownloadSPB`
	`}`
	`-- ASN1STOP`	

2	The SPBM server sends aSI1-63311-response-01 to the service provider client:
	`-- ASN1START`
	`aSI1-63311-response-01 Si1CreateSPReferenceResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Executed-Success`
	`  },`
	`  aSi1CreateSPReferenceResult aSi1CreateSPReferenceOk : {`
	`  aCodeM eCODEM-1,`
	`  aSpbId eSPBID-1`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ1203_010
		RQ1206_095
		RQ1206_097
		RQ1206_103
		RQ1206_105
		RQ1206_106
		RQ1206_107
		RQ1203_011a

12.6.3.3.1.2 SI1_63312 - Si1.CreateSPReference succeed - no CodeM provided

Test ID	SI1_63312
Test objectives	To verify that the service provider client is able to create a reference (CodeM) shared between the service provider client and the SPBM server for a specific SPB ID, and that in case no CodeM is provided as input the SPBM generates the CodeM.
Configuration reference	CSI1_6311
Initial conditions	

The procedure PSI1_6321 shall be successfully executed:
`-- ASN1START`
`aEMPTY_1 OCTET STRING ::= ''H /*<STORE(aEMPTY_1)>*/`
`-- ASN1STOP`	

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

1	The service provider client sends aSI1-63312-command-01 to the SPBM server:
	`-- ASN1START`
	`aSI1-63312-command-01 Si1CreateSPReferenceCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-CREATEREFERENCE-2`
	`  },`
	`  aSpbId eSPBID-4,`
	`  aTaskType eSi1TaskType-DownloadSPB`
	`}`
	`-- ASN1STOP`	

2	The SPBM server sends aSI1-63312-response-01 to the service provider client:
	`-- ASN1START`
	`aSI1-63312-response-01 Si1CreateSPReferenceResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Executed-Success`
	`  },`
	`  aSi1CreateSPReferenceResult aSi1CreateSPReferenceOk : {`
	`  aCodeM '0000000000000000'H, /*<COMPARE(aEMPTY_1,DIF)>*/`
	`  aSpbId eSPBID-4`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ1203_010
		RQ1203_011
		RQ1206_095
		RQ1206_097
		RQ1206_101
		RQ1206_105
		RQ1206_106
		RQ1206_107
		RQ1203_011b

12.6.3.3.1.3 SI1_63313 - Si1.CreateSPReference error - SpbID already linked

Test ID	SI1_63313
Test objectives	To verify that the service provider client fails to create a reference (CodeM) if the SPB ID is already linked to a CodeM.
Configuration reference	CSI1_6311
Initial conditions	
The procedure PSI1_6321 shall be successfully executed.	

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

1	The service provider client sends aSI1-63313-command-01 to the SPBM server:
	`-- ASN1START`
	`aSI1-63313-command-01 Si1CreateSPReferenceCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-CREATEREFERENCE-3`
	`  },`
	`  aSpbId eSPBID-1,`
	`  aTaskType eSi1TaskType-DownloadSPB`
	`}`
	`-- ASN1STOP`	

2	The SPBM server sends aSI1-63313-response-01 to the service provider client:
	`-- ASN1START`
	`aSI1-63313-response-01 Si1CreateSPReferenceResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Failed`
	`  },`
	`  aSi1CreateSPReferenceResult aSi1CreateSPReferenceError :`
	`eSpbIdAlreadyLinked`
	`}`
	`-- ASN1STOP`
		RQ1206_105
		RQ1206_106

12.6.3.3.1.4 SI1_63314 - Si1.CreateSPReference error - SpbID unknown

Test ID	SI1_63324
Test objectives	To verify that the service provider client fails to create a reference (CodeM) if the referenced SPB ID does not exist in the SPBM.

Configuration reference	CSI1_6311	
Initial conditions		
The procedure PSI1_6321 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**

1	The service provider client sends aSI1-63314-command-01 to the SPBM server:
	`-- ASN1START`
	`aSI1-63314-command-01 Si1CreateSPReferenceCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-CREATEREFERENCE-4`
	`  },`
	`  aCodeM eCODEM-3,`
	`  aSpbId eSPBID-UNKNOWN,`
	`  aTaskType eSi1TaskType-DownloadSPB`
	`}`
	`-- ASN1STOP`	

2	The SPBM server sends aSI1-31314-response-01 to the service provider client:
	`-- ASN1START`
	`aSI1-63314-response-01 Si1CreateSPReferenceResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Failed`
	`  },`
	`  aSi1CreateSPReferenceResult aSi1CreateSPReferenceError :`
	`eSpbIdUnknown`
	`}`
	`-- ASN1STOP`
		RQ1206_105
		RQ1206_106

12.6.3.3.1.5 SI1_63315 – Si1.CreateSPReference error - Task type unknown

Test ID	SI1_63315	
Test objectives	To verify that the service provider client fails to create a reference (CodeM) if the Task Type provided as an input is not 'DownloadSPB'.	
Configuration reference	CSI1_6311	
Initial conditions		
The procedure PSI1_6321 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**

1	The service provider client sends aSI1-63315-command-01 to the SPBM server:
	`-- ASN1START`
	`aSI1-63315-command-01 Si1CreateSPReferenceCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-CREATEREFERENCE-5`
	`  },`
	`  aCodeM eCODEM-5,`
	`  aSpbId eSPBID-5,`
	`  aTaskType eSi1TaskType-EgibilityInfo`
	`}`
	`-- ASN1STOP`	

2	The SPBM server sends aSI1-63315-response-01 to the service provider client:
	`-- ASN1START`
	`aSI1-63315-response-01 Si1CreateSPReferenceResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Failed`
	`  },`
	`  aSi1CreateSPReferenceResult aSi1CreateSPReferenceError :`
	`eTaskTypeUnknown`
	`}`
	`-- ASN1STOP`
		RQ1206_099
		RQ1206_105
		RQ1206_106

12.6.3.3.1.6 SI1_63316 – Si1.CreateSPReference error - CodeM not allowed

Test ID	SI1_63316	
Test objectives	To verify that the service provider client fails to create a reference (CodeM) if the CodeM provided as an input is already linked to another SpbID.	
Configuration reference	CSI1_6311	
Initial conditions		
The procedure PSI1_6321 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**

1	The service provider client sends aSI1-63316-command-01 to the SPBM server:
	`-- ASN1START`
	`aSI1-63316-command-01 Si1CreateSPReferenceCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-CREATEREFERENCE-6`
	`  },`
	`  aCodeM eCODEM-1,`
	`  aSpbId eSPBID-2,`
	`  aTaskType eSi1TaskType-DownloadSPB`
	`}`
	`-- ASN1STOP`	

2	The SPBM server sends a SI1-63316-response-01 to the service provider client:
	`-- ASN1START`
	`aSI1-63316-response-01 Si1CreateSPReferenceResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Failed`
	`  },`
	`  aSi1CreateSPReferenceResult aSi1CreateSPReferenceError :`
	`eCodeMNotAllowed`
	`}`
	`-- ASN1STOP`
		RQ1206_102
		RQ1206_105
		RQ1206_106

12.6.3.3.1.7 SI1_63317 – Si1.CreateSPReference error - Task type not allowed

Test identification	SI1_63317	
Test objectives	To verify that the service provider client fails to create a reference (CodeM) if the Task Type provided as an input is not allowed.	
Configuration reference	CSI1_6311	
Initial conditions		
The procedure PSI1_6321 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**

1	The service provider client sends aSI1-63317-command-01 to the SPBM server:
	`-- ASN1START`
	`aSI1-63317-command-01 Si1CreateSPReferenceCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-2,`
	`  aFunctionCallId eFUNCTION-CALL-ID-CREATEREFERENCE-7`
	`  },`
	`  aCodeM eCODEM-6,`
	`  aSpbId eSPBID-6,`
	`  aTaskType eSi1TaskType-DownloadSPB`
	`}`
	`-- ASN1STOP`	

2	The SPBM server sends aSI1-63317-response-01 to the service provider client:
	`-- ASN1START`
	`aSI1-63317-response-01 Si1CreateSPReferenceResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Failed`
	`  },`
	`  aSi1CreateSPReferenceResult aSi1CreateSPReferenceError :`
	`eTaskNotAllowed`
	`}`
	`-- ASN1STOP`
		RQ1206_100
		RQ1206_105
		RQ1206_106

12.6.3.3.2 Si1.SelectSpb command and response handling

12.6.3.3.2.1 SI1_63321 - Si1.SelectSpb succeed

Test ID	SI1_63321	
Test objectives	To verify that the service provider is able to select a SPB in the SPBM.	
Configuration reference	CSI1_6311	
Initial conditions		
The procedure PSI1_6321 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**

1	The service provider client sends aSI1-63321-command-01 to the SPBM server:
	`-- ASN1START`
	`aSI1-63321-command-01 Si1SelectSpbCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-SELECT-1`
	`  },`
	`  aSpbId eSPBID-7,`
	`  aSpbType eSPBTYPE-7,`
	`  aPpIdentifier ePPIDENTIFIER-1,`
	`  aCodeM eCODEM-7,`
	`  aFlagFinalize FALSE`
	`}`
	`-- ASN1STOP`	

2	The SPBM server sends aSI1-63321-response-01 to the service provider client:
	`-- ASN1START`
	`aSI1-63321-response-01 Si1SelectSpbResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Executed-Success`
	`  },`
	`  aSi1SelectSpbResult aSi1SelectSpbOk : {`
	`  aSpbId eSPBID-7,`
	`  aSpbType eSPBTYPE-7,`
	`  aPpIdentifier ePPIDENTIFIER-1,`
	`  aCodeM eCODEM-7`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ1203_008
		RQ1206_080
		RQ1206_081
		RQ1206_082
		RQ1206_088
		RQ1206_089
		RQ1206_090
		RQ1206_092
		RQ1206_093
		RQ1206_094

12.6.3.3.2.2 SI1_63322 - Si1.SelectSpb succeed - CodeM not known

Test ID	SI1_63322
Test objectives	To verify that upon reception of the "Si1.SelectSpb" function command, the SPBM stores the CodeM if provided as input data is not formerly known by the SPBM.
Configuration reference	CSI1_6311
Initial conditions	
The procedure PSI1_6321 shall be successfully executed.	

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

1	The service provider client sends aSI1-63322-command-01 to the SPBM server:
	`-- ASN1START`
	`aSI1-63322-command-01 Si1SelectSpbCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-SELECT-2`
	`  },`
	`  aSpbId eSPBID-2,`
	`  aSpbType eSPBTYPE-2,`
	`  aPpIdentifier ePPIDENTIFIER-1,`
	`  aCodeM eCODEM-NOTKNOWN,`
	`  aFlagFinalize TRUE`
	`}`
	`-- ASN1STOP`	

2	The SPBM server sends aSI1-63322-response-01 to the service provider client:
	`-- ASN1START`
	`aSI1-63322-response-01 Si1SelectSpbResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Executed-Success`
	`  },`
	`  aSi1SelectSpbResult aSi1SelectSpbOk : {`
	`  aSpbId eSPBID-1,`
	`  aSpbType eSPBTYPE-1,`
	`  aPpIdentifier ePPIDENTIFIER-1,`
	`  aCodeM eCODEM-NOTKNOWN`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ1206_080
		RQ1206_081
		RQ1206_082
		RQ1206_087
		RQ1206_088
		RQ1206_089
		RQ1206_090
		RQ1206_092
		RQ1206_093
		RQ1206_094

12.6.3.3.2.3 SI1_63323 - Si1.SelectSpb error - SpbID unknown

Test ID	SI1_63323
Test objectives	To verify that the service provider fails to select a SPB in the SPBM if the selected SpbID does not exist in the SPBM.
Configuration reference	CSI1_6311
Initial conditions	
The procedure PSI1_6321 shall be successfully executed.	

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

1	The service provider client sends aSI1-63323-command-01 to the SPBM server:
	`-- ASN1START`
	`aSI1-63323-command-01 Si1SelectSpbCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-SELECT-3`
	`  },`
	`  aSpbId eSPBID-UNKNOWN,`
	`  aSpbType eSPBTYPE-1,`
	`  aPpIdentifier ePPIDENTIFIER-1,`
	`  aCodeM eCODEM-2,`
	`  aFlagFinalize TRUE`
	`}`
	`-- ASN1STOP`	

2	The SPBM server sends aSI1-63313-response-01 to the service provider client:
	`-- ASN1START`
	`aSI1-63323-response-01 Si1SelectSpbResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Failed`
	`  },`
	`  aSi1SelectSpbResult aSi1SelectSpbError : eSpbIdUnknown`
	`}`
	`-- ASN1STOP`
		RQ1206_084
		RQ1206_092
		RQ1206_093

12.6.3.3.2.4 SI1_63324 - Si1.SelectSpb error - SpbType unknown

Test ID	SI1_63324	
Test objectives	To verify that the service provider fails to select a SPB in the SPBM if the selected SPB Type is unknown.	
Configuration reference	CSI1_6311	
Initial conditions		
The procedure PSI1_6321 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**

1	The service provider client sends aSI1-63324-command-01 to the SPBM server:
	`-- ASN1START`
	`aSI1-63324-command-01 Si1SelectSpbCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-SELECT-4`
	`  },`
	`  aSpbId eSPBID-3,`
	`  aSpbType eSPBTYPE-UNKNOWN,`
	`  aPpIdentifier ePPIDENTIFIER-1,`
	`  aCodeM eCODEM-3,`
	`  aFlagFinalize TRUE`
	`}`
	`-- ASN1STOP`	

2	The SPBM server sends aSI1-63324-response-01 to the service provider client:
	`-- ASN1START`
	`aSI1-63324-response-01 Si1SelectSpbResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Failed`
	`  },`
	`  aSi1SelectSpbResult aSi1SelectSpbError : eSpbTypeUnknown`
	`}`
	`-- ASN1STOP`
		RQ1206_085
		RQ1206_092
		RQ1206_093

12.6.3.3.2.5 SI1_63325 - Si1.SelectSpb error - SpbType mismatch

Test ID	SI1_63325	
Test objectives	To verify that the service provider fails to select a SPB in the SPBM if the selected SPB ID does not match to the selected SPB Type.	
Configuration reference	CSI1_6311	
Initial conditions		
The procedure PSI1_6321 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**

1	The service provider client sends aSI1-63325-command-01 to the SPBM server:
	`-- ASN1START`
	`aSI1-63325-command-01 Si1SelectSpbCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-SELECT-5`
	`  },`
	`  aSpbId eSPBID-1,`
	`  aSpbType eSPBTYPE-2,`
	`  aPpIdentifier ePPIDENTIFIER-1,`
	`  aCodeM eCODEM-1,`
	`  aFlagFinalize TRUE`
	`}`
	`-- ASN1STOP`	

2	The SPBM server sends aSI1-63325-response-01 to the service provider client:
	`-- ASN1START`
	`aSI1-63325-response-01 Si1SelectSpbResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Failed`
	`  },`
	`  aSi1SelectSpbResult aSi1SelectSpbError : eSpbTypeMismatch`
	`}`
	`-- ASN1STOP`
		RQ1206_084
		RQ1206_092
		RQ1206_093

12.6.3.3.2.6 SI1_63326 - Si1.SelectSpb error - CodeM not allowed

Test ID	SI1_63326	
Test objectives	To verify that the service provider fails to select a SPB in the SPBM if the selected CodeM is already linked to a SpbID.	
Configuration reference	CSI1_6311	
Initial conditions		
The procedure PSI1_6321 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**

1	The service provider client sends aSI1-63326-command-01 to the SPBM server:
	`-- ASN1START`
	`aSI1-63326-command-01 Si1SelectSpbCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-SELECT-6`
	`  },`
	`  aSpbId eSPBID-2,`
	`  aSpbType eSPBTYPE-2,`
	`  aPpIdentifier ePPIDENTIFIER-1,`
	`  aCodeM eCODEM-1,`
	`  aFlagFinalize TRUE`
	`}`
	`-- ASN1STOP`	

2	The SPBM server sends aSI1-63326-response-01 to the service provider client:
	`-- ASN1START`
	`aSI1-63326-response-01 Si1SelectSpbResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Failed`
	`  },`
	`  aSi1SelectSpbResult aSi1SelectSpbError : eCodeMNotAllowed`
	`}`
	`-- ASN1STOP`
		RQ1206_086
		RQ1206_092
		RQ1206_093
		RQ1203_008

12.6.3.3.2.7 SI1_63327 - Si1.SelectSpb without FlagFinalize

Test ID	SI1_63327	
Test objectives	To verify that if aFlagFinalize is not present in Si1SelectSpbCommand, it is considered as set to FALSE. The SPBM does not wait for Si1.FinalizePreparation to continue with the Bound SPB image download.	
Configuration reference	CSI1_6312	
Initial conditions		
The procedure PSI1_6321 shall be successfully executed. The procedure PSI1_6322 shall be successfully executed. The aSI2-63327-command-02 is generated by using the SI2_63327_command_02 configuration file. The aSI2-63327-command-03 is generated by using the SI2_63327_command_03 configuration file. The aSI2-63327-response-02 is verified by using the SI2_63327_response_02 configuration file. The aSI2-63327-response-03 is verified by using the SI2_63327_response_03 configuration file.		
Test sequence		
Step	**Description**	**Requirements**

1	The service provider client sends aSI1-63327-command-01 to the SPBM server:
	`-- ASN1START`
	`aSI1-63327-command-01 Si1SelectSpbCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-SELECT-7`
	`  },`
	`  aSpbId eSPBID-1,`
	`  aSpbType eSPBTYPE-1,`
	`  aPpIdentifier ePPIDENTIFIER-1,`
	`  aCodeM eCODEM-1`
	`}`
	`-- ASN1STOP`	

2	The SPBM server sends aSI1-63327-response-01 to the service provider client:
	`-- ASN1START`
	`aSI1-63327-response-01 Si1SelectSpbResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Executed-Success`
	`  },`
	`  aSi1SelectSpbResult aSi1SelectSpbOk : {`
	`  aSpbId eSPBID-1,`
	`  aSpbType eSPBTYPE-1,`
	`  aPpIdentifier ePPIDENTIFIER-1,`
	`  aCodeM eCODEM-1`
	`  }`
	`}`
	`-- ASN1STOP`	

3	The LBA sends aSI2-63327-command-02 Si2.GetSpbmCertificate command the SPBM.	
4	The SPBM sends aSI2-63327-response-02 Si2.GetSpbmCertificate response to the LBA containing aSi2GetSpbmCertificateOk.	RQ1206_091
5	The LBA sends aSI2-63327-command-03 Si2. GetBoundSpbImage command the SPBM.	
6	The SPBM sends aSI2-63327-response-03 Si2. GetBoundSpbImage response to the LBA. The LBA (tester) shall verify that the response is well formatted.	RQ1206_091

12.6.3.3.2.8 SI1_63328 - Si1.SelectSpb with FlagFinalize set to TRUE

Test ID	SI1_63328
Test objectives	To verify that if aFlagFinalize is set to TRUE the SPBM waits for the Si1.FinalizePreparation command to continue with the Bound SPB image download.
Configuration reference	CSI1_6312

Initial conditions

The procedure PSI1_6321 shall be successfully executed.
The procedure PSI1_6322 shall be successfully executed.
The aSI2-63328-command-02 is generated by using the SI2_63328_command_02 configuration file.
The aSI2-63328-command-03 is generated by using the SI2_63328_command_03 configuration file.
The aSI2-63328-response-02 is verified by using the SI2_63328_response_02 configuration file.
The aSI2-63328-response-03 is verified by using the SI2_63328_response_03 configuration file.

Test sequence

Step	Description	Requirements

1	The service provider client sends aSI1-63328-command-01 to the SPBM server:
	`-- ASN1START`
	`aSI1-63328-command-01 Si1SelectSpbCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-SELECT-8`
	`  },`
	`  aSpbId eSPBID-8,`
	`  aSpbType eSPBTYPE-8,`
	`  aPpIdentifier ePPIDENTIFIER-1,`
	`  aCodeM eCODEM-8,`
	`  aFlagFinalize TRUE`
	`}`
	`-- ASN1STOP`	

2	The SPBM server sends aSI1-63328-response-01 to the service provider client:
	`-- ASN1START`
	`aSI1-63328-response-01 Si1SelectSpbResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Executed-Success`
	`  },`
	`  aSi1SelectSpbResult aSi1SelectSpbOk : {`
	`  aSpbId eSPBID-8,`
	`  aSpbType eSPBTYPE-8,`
	`  aPpIdentifier ePPIDENTIFIER-1,`
	`  aCodeM eCODEM-8`
	`  }`
	`}`
	`-- ASN1STOP`	

3	The service provider client sends aSI1-63328-command-02 to the SPBM server:
	`-- ASN1START`
	`aSI1-63328-command-02 Si1FinalizePreparationCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-FINALIZEPREP-1`
	`  },`
	`  aCodeM eCODEM-8`
	`}`
	`-- ASN1STOP`	

4	The SPBM server sends aSI1-63328-response-02 to the service provider client:
	`-- ASN1START`
	`aSI1-63328-response-02 Si1FinalizePreparationResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Executed-Success`
	`  },`
	`  aSi1FinalizePreparationResult aSi1FinalizePreparationOk : {`
	`  aCodeM eCODEM-8`
	`  }`
	`}`
	`-- ASN1STOP`	

5	The LBA sends aSI2-63328-command-03 Si2.GetSpbmCertificate command the SPBM.	
6	The SPBM sends aSI2-63328-response-03 Si2.GetSpbmCertificate response to the LBA containing aSi2GetSpbmCertificateOk.	RQ1206_110 RQ1206_118
7	The LBA sends aSI2-63328-command-04 Si2. GetBoundSpbImage command the SPBM.	

8	The SPBM sends aSI2-63328-response-04 Si2. GetBoundSpbImage response to the LBA. The LBA (tester) shall verify that the response is well formatted.	RQ1206_110 RQ1206_118 RQ1203_009

12.6.3.3.3 Si1.FinalizePreparation command and response handling

12.6.3.3.3.1 SI1_63331 - Si1.FinalizePreparation succeed

Test ID	SI1_63331
Test objectives	Check if the service provider uses the "Si1.FinalizePreparation" function to indicate that its internal procedures are completed.
Configuration reference	CSI1_6311
Initial conditions	
The procedure PSI1_6321 shall be successfully executed.	
Test sequence	

Step	Description	Requirements

1	The service provider client sends aSI1-63331-command-01 to the SPBM server:
	`-- ASN1START`
	`aSI1-63331-command-01 Si1SelectSpbCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-SELECT-9`
	`  },`
	`  aSpbId eSPBID-1,`
	`  aSpbType eSPBTYPE-1,`
	`  aPpIdentifier ePPIDENTIFIER-1,`
	`  aCodeM eCODEM-1,`
	`  aFlagFinalize TRUE`
	`}`
	`-- ASN1STOP`	

2	The SPBM server sends aSI1-63331-response-01 to the service provider client:
	`-- ASN1START`
	`aSI1-63331-response-01 Si1SelectSpbResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Executed-Success`
	`  },`
	`  aSi1SelectSpbResult aSi1SelectSpbOk : {`
	`  aSpbId eSPBID-1,`
	`  aSpbType eSPBTYPE-1,`
	`  aPpIdentifier ePPIDENTIFIER-1,`
	`  aCodeM eCODEM-1`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ1206_108
		RQ1206_109

3	The service provider client sends SI1-63331-command-02 to the SPBM server:
	`-- ASN1START`
	`aSI1-Y331-command-02 Si1FinalizePreparationCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-FINALIZEPREP-2`
	`  },`
	`  aCodeM eCODEM-1`
	`}`
	`-- ASN1STOP`	

4	The SPBM server sends aSI1-63331-response-02 to the service provider client:
	`-- ASN1START`
	`aSI1-63331-response-02 Si1FinalizePreparationResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Executed-Success`
	`  },`
	`  aSi1FinalizePreparationResult aSi1FinalizePreparationOk : {`
	`  aCodeM eCODEM-1`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ1203_009
		RQ1206_108
		RQ1206_109
		RQ1206_111
		RQ1206_113
		RQ1206_116
		RQ1206_117
		RQ1206_119

12.6.3.3.3.2 SI1_63332 - Si1.FinalizePreparation error - CodeM unknown

Test ID	SI1_63332	
Test objectives	Verify that the "Si1.FinalizePreparation" function fails, if the CodeM provided as input is unknown.	
Configuration reference	CSI1_6311	
Initial conditions		
The procedure PSI1_6321 shall be successfully executed.		
Test sequence		
Step	Description	Requirements

1	The service provider client sends aSI1-63332-command-01 to the SPBM server:
	`-- ASN1START`
	`aSI1-63332-command-01 Si1SelectSpbCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-SELECT-10`
	`  },`
	`  aSpbId eSPBID-1,`
	`  aSpbType eSPBTYPE-1,`
	`  aPpIdentifier ePPIDENTIFIER-1,`
	`  aCodeM eCODEM-1,`
	`  aFlagFinalize TRUE`
	`}`
	`-- ASN1STOP`	

2	The SPBM server sends aSI1-63332-response-01 to the service provider client:
	`-- ASN1START`
	`aSI1-63332-response-01 Si1SelectSpbResponse ::= {`
	`  aS1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Executed-Success`
	`  },`
	`  aSi1SelectSpbResult aSi1SelectSpbOk : {`
	`  aSpbId eSPBID-1,`
	`  aSpbType eSPBTYPE-1,`
	`  aPpIdentifier ePPIDENTIFIER-1,`
	`  aCodeM eCODEM-1`
	`  }`
	`}`
	`-- ASN1STOP`	

3	The service provider client sends aSI1-63332-command-02 to the SPBM server:
	`-- ASN1START`
	`aSI1-63332-command-02 Si1FinalizePreparationCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-FINALIZEPREP-3`
	`  },`
	`  aCodeM eCODEM-UNKNOWN`
	`}`
	`-- ASN1STOP`	

4	The SPBM server sends aSI1-63332-response-02 to the service provider client:
	`-- ASN1START`
	`aSI1-63332-response-02 Si1FinalizePreparationResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Failed`
	`  },`
	`  aSi1FinalizePreparationResult aSi1FinalizePreparationError :`
	`eCodeMUnknown`
	`}`
	`-- ASN1STOP`
		RQ1206_113
		RQ1206_114
		RQ1206_116
		RQ1206_117
		RQ1206_119

12.6.3.3.3.3 SI1_63333 - Si1.FinalizePreparation error - CodeM unlinked

Test ID	SI1_63333
Test objectives	Verify that the "Si1.FinalizePreparation" function fails, if the CodeM provided as input is not linked to any SPB.
Configuration reference	CSI1_6311
Initial conditions	
The procedure PSI1_6321 shall be successfully executed.	

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

1	The service provider client sends aSI1-63333-command-01 to the SPBM server:
	`-- ASN1START`
	`aSI1-63333-command-01 Si1SelectSpbCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-SELECT-11`
	`  },`
	`  aSpbId eSPBID-1,`
	`  aSpbType eSPBTYPE-1,`
	`  aPpIdentifier ePPIDENTIFIER-1,`
	`  aCodeM eCODEM-1,`
	`  aFlagFinalize TRUE`
	`}`
	`-- ASN1STOP`	

2	The SPBM server sends aSI1-63333-response-01 to the service provider client:
	`-- ASN1START`
	`aSI1-63333-response-01 Si1SelectSpbResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Executed-Success`
	`  },`
	`  aSi1SelectSpbResult aSi1SelectSpbOk : {`
	`  aSpbId eSPBID-1,`
	`  aSpbType eSPBTYPE-1,`
	`  aPpIdentifier ePPIDENTIFIER-1,`
	`  aCodeM eCODEM-1`
	`  }`
	`}`
	`-- ASN1STOP`	

3	The service provider client sends aSI1-63333-command-02 to the SPBM server:
	`-- ASN1START`
	`aSI1-63333-command-02 Si1FinalizePreparationCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-FINALIZEPREP-4`
	`  },`
	`  aCodeM eCODEM-UNLINKED`
	`}`
	`-- ASN1STOP`	

4	The SPBM server sends aSI1-63333-response-02 to the service provider client:
	`-- ASN1START`
	`aSI1-63333-response-02 Si1FinalizePreparationResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Failed`
	`  },`
	`  aSi1FinalizePreparationResult aSi1FinalizePreparationError :`
	`eCodeMNotAllowed`
	`}`
	`-- ASN1STOP`
		RQ1206_113
		RQ1206_115
		RQ1206_116
		RQ1206_117
		RQ1206_119

12.6.3.3.4 Si1.CancelPreparation command and response handling

12.6.3.3.4.1 SI1_63341 - Si1.CancelPreparation succeed with SpbID

Test ID	SI1_63341
Test objectives	Check if the service provider uses the "Si1.CancelPreparation" function to cancel a pending preparation procedure.
Configuration reference	CSI1_6311
Initial conditions	
The procedure PSI1_6321 shall be successfully executed.	
Test sequence	
Step	**Description**	**Requirements**

1	The service provider client sends aSI1-63341-command-01 to the SPBM server:
	`-- ASN1START`
	`aSI1-63341-command-01 Si1SelectSpbCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-SELECT-12`
	`  },`
	`  aSpbId eSPBID-1,`
	`  aSpbType eSPBTYPE-1,`
	`  aPpIdentifier ePPIDENTIFIER-1,`
	`  aCodeM eCODEM-1,`
	`  aFlagFinalize TRUE`
	`}`
	`-- ASN1STOP`	

2	The SPBM server sends aSI1-63341-response-01 to the service provider client:
	`-- ASN1START`
	`aSI1-63341-response-01 Si1SelectSpbResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Executed-Success`
	`  },`
	`  aSi1SelectSpbResult aSi1SelectSpbOk : {`
	`  aSpbId eSPBID-1,`
	`  aSpbType eSPBTYPE-1,`
	`  aPpIdentifier ePPIDENTIFIER-1,`
	`  aCodeM eCODEM-1`
	`  }`
	`}`
	`-- ASN1STOP`	

3	The service provider client sends aSI1-63341-command-02 to the SPBM server:
	`-- ASN1START`
	`aSI1-63341-command-02 Si1CancelPreparationCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-CANCELPREP-1`
	`  },`
	`  aSpbId eSPBID-1`
	`}`
	`-- ASN1STOP`	

4	The SPBM server sends aSI1-63341-response-02 to the service provider client:
	`-- ASN1START`
	`aSI1-63341-response-02 Si1CancelPreparationResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Executed-Success`
	`  },`
	`  aSi1CancelPreparationResult aSi1CancelPreparationOk : {`
	`  aSpbId eSPBID-1`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ1206_120
		RQ1206_121
		RQ1206_130
		RQ1206_131
		RQ1206_132

12.6.3.3.4.2 SI1_63342 - Si1.CancelPreparation succeed with CodeM

Test ID	SI1_63342
Test objectives	Check if the service provider uses the "Si1.CancelPreparation" function to cancel a pending preparation procedure.
Configuration reference	CSI1_6211
Initial conditions	
The procedure PSI1_6321 shall be successfully executed.	

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

1	The service provider client sends aSI1-63342-command-01 to the SPBM server:
	`-- ASN1START`
	`aSI1-63342-command-01 Si1SelectSpbCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-SELECT-13`
	`  },`
	`  aSpbId eSPBID-1,`
	`  aSpbType eSPBTYPE-1,`
	`  aPpIdentifier ePPIDENTIFIER-1,`
	`  aCodeM eCODEM-1,`
	`  aFlagFinalize TRUE`
	`}`
	`-- ASN1STOP`	

2	The SPBM server sends aSI1-63342-response-01 to the service provider client:
	`-- ASN1START`
	`aSI1-63342-response-01 Si1SelectSpbResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Executed-Success`
	`  },`
	`  aSi1SelectSpbResult aSi1SelectSpbOk : {`
	`  aSpbId eSPBID-1,`
	`  aSpbType eSPBTYPE-1,`
	`  aPpIdentifier ePPIDENTIFIER-1,`
	`  aCodeM eCODEM-1`
	`  }`
	`}`
	`-- ASN1STOP`	

3	The service provider client sends aSI1-63342-command-02 to the SPBM server:
	`-- ASN1START`
	`aSI1-63342-command-02 Si1CancelPreparationCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-CANCELPREP-2`
	`  },`
	`  aCodeM eCODEM-1`
	`}`
	`-- ASN1STOP`	

4	The SPBM server sends aSI1-63342-response-02 to the service provider client:
	`-- ASN1START`
	`SI1-63342-response-02 Si1CancelPreparationResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Executed-Success`
	`  },`
	`  aSi1CancelPreparationResult aSi1CancelPreparationOk : {`
	`  aCodeM eCODEM-1`
	`  }`
	`-- ASN1STOP`
		RQ1203_012
		RQ1206_120
		RQ1206_121
		RQ1206_130
		RQ1206_131
		RQ1206_132

5	The service provider client sends aSI1-63342-command-03 to the SPBM server:
	`-- ASN1START`
	`aSI1-63342-command-03 Si1CreateSPReferenceCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-CREATEREFERENCE-8`
	`  },`
	`  aCodeM eCODEM-1,`
	`  aSpbId eSPBID-1,`
	`  aTaskType eSi1TaskType-DownloadSPB`
	`}`
	`-- ASN1STOP`	

6	The SPBM server sends aSI1-63342-response-03 to the service provider client:
	`-- ASN1START`
	`aSI1-63342-response-03 Si1CreateSPReferenceResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Executed-Success`
	`  },`
	`  aSi1CreateSPReferenceResult aSi1CreateSPReferenceOk : {`
	`  aCodeM eCODEM-1,`
	`  aSpbId eSPBID-1`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ1206_127
		RQ1206_128
		RQ1206_129

7	The service provider client sends aSI1-63342-command-04 to the SPBM server:
	`-- ASN1START`
	`aSI1-63342-command-04 Si1CancelPreparationCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-CANCELPREP-3`
	`  },`
	`  aCodeM eCODEM-1`
	`}`
	`-- ASN1STOP`	

8	The SPBM server sends aSI1-63342-response-04 to the service provider client:
	`-- ASN1START`
	`aSI1-63342-response-04 Si1CancelPreparationResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Executed-Success`
	`  },`
	`  aSi1CancelPreparationResult aSi1CancelPreparationOk : {`
	`  aCodeM eCODEM-1`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ1203_012

12.6.3.3.4.3 SI1_63343 - Si1.CancelPreparation succeed with CodeM and SpbID

Test ID	SI1_63343
Test objectives	Check if the service provider uses the "Si1.CancelPreparation" function to cancel a pending preparation procedure.
Configuration reference	CSI1_6311
Initial conditions	
The procedure PSI1_6321 shall be successfully executed.	
Test sequence	

Step	**Description**	**Requirements**

1	The service provider client sends aSI1-63343-command-01 to the SPBM server:
	`-- ASN1START`
	`aSI1-63343-command-01 Si1SelectSpbCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-SELECT-14`
	`  },`
	`  aSpbId eSPBID-1,`
	`  aSpbType eSPBTYPE-1,`
	`  aPpIdentifier ePPIDENTIFIER-1,`
	`  aCodeM eCODEM-1,`
	`  aFlagFinalize TRUE`
	`}`
	`-- ASN1STOP`	

2	The SPBM server sends aSI1-63343-response-01 to the service provider client:
	`-- ASN1START`
	`aSI1-63343-response-01 Si1SelectSpbResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Executed-Success`
	`  },`
	`  aSi1SelectSpbResult aSi1SelectSpbOk : {`
	`  aSpbId eSPBID-1,`
	`  aSpbType eSPBTYPE-1,`
	`  aPpIdentifier ePPIDENTIFIER-1,`
	`  aCodeM eCODEM-1`
	`  }`
	`}`
	`-- ASN1STOP`	

3	The service provider client sends aSI1-63343-command-02 to the SPBM server:
	`-- ASN1START`
	`aSI1-63343-command-02 Si1CancelPreparationCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-CANCELPREP-4`
	`  },`
	`  aCodeM eCODEM-1,`
	`  aSpbId eSPBID-1`
	`}`
	`-- ASN1STOP`	

4	The SPBM server sends aSI1-63343-response-02 to the service provider client:
	`-- ASN1START`
	`aSI1-63343-response-02 Si1CancelPreparationResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Executed-Success`
	`  },`
	`  aSi1CancelPreparationResult aSi1CancelPreparationOk : {`
	`  aCodeM eCODEM-1,`
	`  aSpbId eSPBID-1`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ1203_012
		RQ1206_120
		RQ1206_121
		RQ1206_130
		RQ1206_131
		RQ1206_132

5	The service provider client sends aSI1-63343-command-03 to the SPBM server:
	`-- ASN1START`
	`aSI1-63343-command-03 Si1CreateSPReferenceCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-CREATEREFERENCE-10`
	`  },`
	`  aCodeM eCODEM-1,`
	`  aSpbId eSPBID-1,`
	`  aTaskType eSi1TaskType-DownloadSPB`
	`}`
	`-- ASN1STOP`	

6	The SPBM server sends aSI1-63343-response-03 to the service provider client:
	`-- ASN1START`
	`aSI1-63343-response-03 Si1CreateSPReferenceResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Executed-Success`
	`  },`
	`  aSi1CreateSPReferenceResult aSi1CreateSPReferenceOk : {`
	`  aCodeM eCODEM-1,`
	`  aSpbId eSPBID-1`
	`  }`
	`}`
	`-- ASN1STOP`
		RQ1206_127
		RQ1206_128
		RQ1206_129

7	The service provider client sends aSI1-63343-command-04 to the SPBM server:
	`-- ASN1START`
	`aSI1-63343-command-04 Si1CancelPreparationCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-CANCELPREP-5`
	`  },`
	`  aCodeM eCODEM-1`
	`}`
	`-- ASN1STOP`	

8	The SPBM server sends aSI1-63343-response-04 to the service provider client:
	`-- ASN1START`
	`aSI1-63343-response-04 Si1CancelPreparationResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Executed-Success`
	`  },`
	`  aSi1CancelPreparationResult aSi1CancelPreparationOk : {`
	`  aCodeM eCODEM-1`
	`  }`
	`}`
	`-- ASN1STOP`	

12.6.3.3.4.4 SI1_63344 - Si1.CancelPreparation error - CodeM unknown

Test ID	SI1_63344	
Test objectives	Verify that the "Si1.CancelPreparation" function fails, if the CodeM provided as input is unknown.	
Configuration reference	CSI1_6311	
Initial conditions		
The procedure PSI1_6321 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**

1	The service provider client sends aSI1-63344-command-01 to the SPBM server:
	`-- ASN1START`
	`aSI1-63344-command-01 Si1SelectSpbCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-SELECT-15`
	`  },`
	`  aSpbId eSPBID-1,`
	`  aSpbType eSPBTYPE-1,`
	`  aPpIdentifier ePPIDENTIFIER-1,`
	`  aCodeM eCODEM-1,`
	`  aFlagFinalize TRUE`
	`}`
	`-- ASN1STOP`	

2	The SPBM server sends aSI1-63344-response-01 to the service provider client:
	`-- ASN1START`
	`aSI1-63344-response-01 Si1SelectSpbResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Executed-Success`
	`  },`
	`  aSi1SelectSpbResult aSi1SelectSpbOk : {`
	`  aSpbId eSPBID-1,`
	`  aSpbType eSPBTYPE-1,`
	`  aPpIdentifier ePPIDENTIFIER-1,`
	`  aCodeM eCODEM-1`
	`  }`
	`}`
	`-- ASN1STOP`	

3	The service provider client sends aSI1-63344-command-02 to the SPBM server:
	`-- ASN1START`
	`aSI1-63344-command-02 Si1CancelPreparationCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-CANCELPREP-6`
	`  },`
	`  aCodeM eCODEM-UNKNOWN`
	`}`
	`-- ASN1STOP`	

4	The SPBM server sends aSI1-63344-response-02 to the service provider client:
	`-- ASN1START`
	`aSI1-63344-response-02 Si1CancelPreparationResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Failed`
	`  },`
	`  aSi1CancelPreparationResult aSi1CancelPreparationError :`
	`eCodeMUnknown`
	`}-- ASN1STOP`
		RQ1206_123
		RQ1206_130
		RQ1206_131
		RQ1206_132

12.6.3.3.4.5 SI1_63345 - Si1.CancelPreparation error - SpbID unknown

Test ID	SI1_63345	
Test objectives	Verify that the "Si1.CancelPreparation" function fails, if the SpbID provided as input is unknown.	
Configuration reference	CSI1_6311	
Initial conditions		
The procedure PSI1_6321 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**

1	The service provider client sends aSI1-63345-command-01 to the SPBM server:
	`-- ASN1START`
	`aSI1-63345-command-01 Si1SelectSpbCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-SELECT-16`
	`  },`
	`  aSpbId eSPBID-1,`
	`  aSpbType eSPBTYPE-1,`
	`  aPpIdentifier ePPIDENTIFIER-1,`
	`  aCodeM eCODEM-1,`
	`  aFlagFinalize TRUE`
	`}`
	`-- ASN1STOP`	

2	The SPBM server sends aSI1-63345-response-01 to the service provider client:
	`-- ASN1START`
	`aSI1-63345-response-01 Si1SelectSpbResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Executed-Success`
	`  },`
	`  aSi1SelectSpbResult aSi1SelectSpbOk : {`
	`  aSpbId eSPBID-1,`
	`  aSpbType eSPBTYPE-1,`
	`  aPpIdentifier ePPIDENTIFIER-1,`
	`  aCodeM eCODEM-1`
	`  }`
	`}`
	`-- ASN1STOP`	

3	The service provider client sends aSI1-63345-command-02 to the SPBM server:
	`-- ASN1START`
	`aSI1-63345-command-02 Si1CancelPreparationCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-CANCELPREP-7`
	`  },`
	`  aSpbId eSPBID-UNKNOWN`
	`}`
	`-- ASN1STOP`	

4	The SPBM server sends aSI1-63345-response-02 to the service provider client:
	`-- ASN1START`
	`aSI1-63345-response-02 Si1CancelPreparationResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Failed`
	`  },`
	`  aSi1CancelPreparationResult aSi1CancelPreparationError :`
	`eSpbIdUnknown`
	`}-- ASN1STOP`
		RQ1206_125
		RQ1206_130
		RQ1206_131
		RQ1206_132

12.6.3.3.4.6 SI1_63346 - Si1.CancelPreparation error - SpbID not allowed

Test ID	SI1_63346	
Test objectives	Verify that the"Si1.CancelPreparation" function fails, if the SpbId provided as input data is not linked to the CodeM provided as input data.	
Configuration reference	CSI1_6311	
Initial conditions		
The procedure PSI1_6321 shall be successfully executed.		
Test sequence		
Step	Description	Requirements

1	The service provider client sends aSI1-63346-command-01 to the SPBM server:
	`-- ASN1START`
	`aSI1-63346-command-01 Si1SelectSpbCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-SELECT-17`
	`  },`
	`  aSpbId eSPBID-1,`
	`  aSpbType eSPBTYPE-1,`
	`  aPpIdentifier ePPIDENTIFIER-1,`
	`  aCodeM eCODEM-1,`
	`  aFlagFinalize TRUE`
	`}`
	`-- ASN1STOP`	

2	The SPBM server sends aSI1-63346-response-01 to the service provider client:
	`-- ASN1START`
	`aSI1-63346-response-01 Si1SelectSpbResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Executed-Success`
	`  },`
	`  aSi1SelectSpbResult aSi1SelectSpbOk : {`
	`  aSpbId eSPBID-1,`
	`  aSpbType eSPBTYPE-1,`
	`  aPpIdentifier ePPIDENTIFIER-1,`
	`  aCodeM eCODEM-1`
	`  }`
	`}`
	`-- ASN1STOP`	

3	The service provider client sends aSI1-63346-command-02 to the SPBM server:
	`-- ASN1START`
	`aSI1-63346-command-02 Si1CancelPreparationCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-CANCELPREP-8`
	`  },`
	`  aCodeM eCODEM-3,`
	`  aSpbId eSPBID-1`
	`}`
	`-- ASN1STOP`	

4	The SPBM server sends aSI1-63346-response-02 to the service provider client:
	`-- ASN1START`
	`aSI1-63346-response-02 Si1CancelPreparationResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Failed`
	`  },`
	`  aSi1CancelPreparationResult aSi1CancelPreparationError :`
	`eSpbIdNotAllowed`
	`}-- ASN1STOP`
		RQ1206_126
		RQ1206_130
		RQ1206_131
		RQ1206_132

12.6.3.3.4.7 SI1_63347 - Si1.CancelPreparation error - CodeM not allowed

Test ID	SI1_63347
Test objectives	Verify that the "Si1.CancelPreparation" function fails if the bound SPB image download procedure associated with the Secondary Platform Bundle identifier linked to the CodeM provided as input data is completed.
Configuration reference	CSI1_6312

Initial conditions

The procedure PSI1_6321 shall be successfully executed.
The procedure PSI1_6322 shall be successfully executed.
The aSI2-63347-command-03 is generated by using the SI2_63347_command_03 configuration file.
The aSI2-63347-command-04 is generated by using the SI2_63347_command_04 configuration file.
The aSI2-63347-response-03 is verified by using the SI2_63347_response_03 configuration file.
The aSI2-63347-response-04 is verified by using the SI2_63347_response_04 configuration file.

Test sequence

Step	Description	Requirements

1	The service provider client sends aSI1-63347-command-01 to the SPBM server:
	`-- ASN1START`
	`aSI1-63347-command-01 Si1SelectSpbCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-SELECT-18`
	`  },`
	`  aSpbId eSPBID-9,`
	`  aSpbType eSPBTYPE-9,`
	`  aPpIdentifier ePPIDENTIFIER-1,`
	`  aCodeM eCODEM-9,`
	`  aFlagFinalize TRUE`
	`}`
	`-- ASN1STOP`	

2	The SPBM server sends aSi1SelectSpbResponse to the service provider client:
	`-- ASN1START`
	`aSI1-63347-response-01 Si1SelectSpbResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Executed-Success`
	`  },`
	`  aSi1SelectSpbResult aSi1SelectSpbOk : {`
	`  aSpbId eSPBID-9,`
	`  aSpbType eSPBTYPE-9,`
	`  aPpIdentifier ePPIDENTIFIER-1,`
	`  aCodeM eCODEM-9`
	`  }`
	`}`
	`-- ASN1STOP`	

3	The service provider client sends aSI1-63347-command-02 to the SPBM server:
	`-- ASN1START`
	`aSI1-63347-command-02 Si1FinalizePreparationCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-FINALIZEPREP-5`
	`  },`
	`  aCodeM eCODEM-9`
	`}`
	`-- ASN1STOP`	

4	The SPBM server sends aSI1-63347-response-02 to the service provider client:
	`-- ASN1START`
	`aSI1-63347-response-02 Si1FinalizePreparationResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Executed-Success`
	`  },`
	`  aSi1FinalizePreparationResult aSi1FinalizePreparationOk : {`
	`  aCodeM eCODEM-9`
	`  }`
	`}`
	`-- ASN1STOP`	

5	The LBA sends aSI2-63347-command-03 Si2.GetSpbmCertificate command the SPBM.	
6	The SPBM sends aSI2-63347-response-03 Si2.GetSpbmCertificate response to the LBA containing aSi2GetSpbmCertificateOk.	

7	The LBA sends aSI2-63347-command-04 Si2.GetBoundSpbImage command the SPBM.	
8	The SPBM sends aSI2-63347-response-04 Si2.GetBoundSpbImage response to the LBA. The LBA (tester) shall verify that the response is well formatted.	

9	The service provider client sends aSI1-63347-command-05 to the SPBM server:
	`-- ASN1START`
	`aSI1-63347-command-05 Si1CancelPreparationCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-CANCELPREP-9`
	`  },`
	`  aCodeM eCODEM-9,`
	`  aSpbId eSPBID-9`
	`}`
	`-- ASN1STOP`	

10	The SPBM server sends aSI1-63347-response-05 to the service provider client:
	`-- ASN1START`
	`aSI1-63347-response-05 Si1CancelPreparationResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Failed`
	`  },`
	`  aSi1CancelPreparationResult aSi1CancelPreparationError :`
	`eCodeMNotAllowed`
	`}-- ASN1STOP`
		RQ1206_124

12.6.3.3.5 Si1.HandleNotification command handling

12.6.3.3.5.1 SI1_63351 - Si1.HandleNotification

Test ID	SI1_63351
Test objectives	Verify that the SPBM is able to send a notification to the service provider.
Configuration reference	CSI1_6312

Initial conditions

The procedure PSI1_6321 shall be successfully executed.
The procedure PSI1_6322 shall be successfully executed.
The aSI2-63351-command-03 is generated by using the SI2_63351_command_03 configuration file.
The aSI2-63351-command-04 is generated by using the SI2_63351_command_04 configuration file.
The aSI2-63351-command-05 is generated by using the SI2_63351_command_05 configuration file.
The aSI2-63351-response-03 is verified by using the SI2_63351_response_03 configuration file.
The aSI2-63351-response-04 is verified by using the SI2_63351_response_04 configuration file.
The aSI2-63351-response-05 is verified by using the SI2_63351_response_05 configuration file.

```
-- ASN1START
aEMPTY_2 OCTET STRING ::= ''H /*<STORE(aEMPTY_2)>*/
aTIME GeneralizedTime ::= "20000101000000.000" /*<STORE(aTIME)>*/
-- ASN1STOP
```

Test sequence

Step	Description	Requirements

1	The service provider client sends aSI1-63351-command-01 to the SPBM server:
	`-- ASN1START`
	`aSI1-63351-command-01 Si1SelectSpbCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-SELECT-19`
	`  },`
	`  aSpbId eSPBID-10,`
	`  aSpbType eSPBTYPE-10,`
	`  aPpIdentifier ePPIDENTIFIER-1,`
	`  aCodeM eCODEM-10,`
	`  aFlagFinalize TRUE`
	`}`
	`-- ASN1STOP`	

2	The SPBM server sends aSI1-63351-response-01 to the service provider client:
	`-- ASN1START`
	`aSI1-63351-response-01 Si1SelectSpbResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Executed-Success`
	`  },`
	`  aSi1SelectSpbResult aSi1SelectSpbOk : {`
	`  aSpbId eSPBID-10,`
	`  aSpbType eSPBTYPE-10,`
	`  aPpIdentifier ePPIDENTIFIER-1,`
	`  aCodeM eCODEM-10`
	`  }`
	`}`
	`-- ASN1STOP`	

3	The service provider client sends aSI1-63351-command-02 to the SPBM server:
	`-- ASN1START`
	`aSI1-63351-command-02 Si1FinalizePrepationCommand ::= {`
	`  aSi1CommandHeader {`
	`  aFunctionRequesterId eFUNCTION-REQUESTER-ID-1,`
	`  aFunctionCallId eFUNCTION-CALL-ID-FINALIZEPREP-6`
	`  },`
	`  aCodeM eCODEM-10`
	`}`
	`-- ASN1STOP`	

4	The SPBM server sends aSI1-63351-response-02 to the service provider client:
	`-- ASN1START`
	`aSI1-63351-response-02 Si1FinalizePreparationResponse ::= {`
	`  aSi1ResponseHeader {`
	`  aFunctionExecutionStatus eSi1ExecutionStatus-Executed-Success`
	`  },`
	`  aSi1FinalizePreparationResult aSi1FinalizePreparationOk : {`
	`  aCodeM eCODEM-10`
	`  }`
	`}`
	`-- ASN1STOP`	

5	The LBA sends aSI2-63351-command-03 Si2.GetSpbmCertificate command to the SPBM.	
6	The SPBM sends aSI2-63351-response-03 Si2.GetSpbmCertificate response to the LBA containing aSi2GetSpbmCertificateOk.	
7	The LBA sends aSI2-63351-command-04 Si2.GetBoundSpbImage command to the SPBM.	
8	The SPBM sends aSI2-63351-response-04 Si2.GetBoundSpbImage response to the LBA. The LBA (tester) shall verify that the response is well formatted.	
9	The LBA sends aSI2-63351-command-05 Si2.HandleNotification command with eNotificationStatus_SPBdownload to the SPBM.	
10	The SPBM sends aSI2-63351-response-05 Si2.HandleNotification response to the LBA.	

11	The SPBM sends aSI1-63351-command-06 Si1.HandleNotification command with eNotificationStatus_SPBdownload to the service provider client:
	`-- ASN1START`
	`aSI1-63351-command-03 Si1HandleNotificationBlock ::= {`
	`  aHandleNotificationHeader {`
	`  aNotificationReceiverId eFUNCTION-REQUESTER-ID-1,`
	`  aNotificationCallId '00000000'H /* <COMPARE(aEMPTY_2,DIF)>*/,`
	`  },`
	`  aCodeM eCODEM-10,`
	`  aSpbId eSPBID-10,`
	`  aSpbType eSPBTYPE-10,`
	`  aPpIdentifier ePPIDENTIFIER-1,`
	`  aTimeStamp "20000101000000.000" /* <COMPARE(aTIME,EQ,DIF)>*/,`
	`  aNotificationEvent eNotificationStatus_SPBDownload,`
	`  aNotificationEventStatus eSi1ExecutionStatus-Executed-Success`
	`}`
	`-- ASN1STOP`
		RQ1205_001
		RQ1206_002
		RQ1206_133
		RQ1206_134
		RQ1206_135

12.6.3.4 Requirements not testable, implicitly verified or verified elsewhere

The following requirements 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:

RQ1206_001, RQ1206_072, RQ1206_073, RQ1206_074, RQ1206_075, RQ1206_076, RQ1206_077, RQ1206_078, RQ1206_079, RQ1206_081 and RQ1206_096.

The following requirements are generated from descriptive text. A verification by tests defined within the present document is not possible:

RQ1206_083, RQ1206_098, RQ1206_104, RQ1206_112, RQ1206_122.

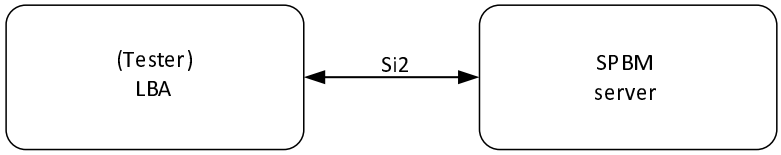
12.6.3.5 ASN.1 Stop

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

12.6.4 Si2 interface

12.6.4.1 Configurations

12.6.4.1.1 CSI2_6411 - SPBM-LBA (tester)

Configuration ID	CSI2_6411
Configuration description	

12.6.4.2 Procedures

12.6.4.2.1 PSI2_6421 - session opening between LBA and the SBPM

Procedure ID		PSI2_6421
Procedure objectives		To put the LBA in charge of managing the connection establishment to the SPBM for the Si2 interface. The binding of the Si2 interface shall be performed over Hypertext Transfer Protocol version 2 (HTTP/2) as defined in IETF RFC 7540 [26] and the Transport Layer Security (TLS) version 1.3 or higher in mutual authentication mode as defined in IETF RFC 8446 [27].
Configuration reference		CSI2_6411
Initial conditions		
Test sequence		
Step	**Description**	
1	The LBA establishes the Si2 session.	
2	The SPBM accepts the Si2 connection.	

12.6.4.3 Test descriptions

12.6.4.3.1 Si2.GetSpbmCertificate command and response handling

12.6.4.3.1.1 SI2_64311 - Si2.GetSpbmCertificate request – normal process

Test ID	SI2_64311	
Test objectives	To verify that the LBA is able to request the SPBM certificates by sending the Si2.GetSpbmCertificate command to the SPBM. To verify that the SPBM sends a response which includes aSi2GetSpbmCertificateOk.	
Configuration reference	CSI2_6411	
Initial conditions		
The SI2_111_command_01 is generated by using the SI2_111_command configuration file. The SI2_111_response_01 is verified by using the SI2_111_response configuration file. The procedure PSI2_6421 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**
1	The LBA sends SI2_111_command_01 to the SPBM.	
2	The SPBM sends SI2_111_response_01 to the LBA containing aSi2.GetSpbmCertificateOk.	RQ1206_131 RQ1206_140 RQ1206_141 RQ1206_142 RQ1206_143 RQ1206_144 RQ1206_145 RQ1206_146 RQ1206_147 RQ1206_148 RQ1206_149 RQ1206_150 RQ1206_152 RQ1206_153 RQ1206_154

12.6.4.3.1.2 SI2_64312 - Si2.GetSpbmCertificate response

Test ID	SI2_64312	
Test objectives	To verify that the SPBM is able to verify the Si2.GetSpbmCertificate response from the LBA. To verify that the Si2.GetSpbmCertificate response is well formatted and handled successfully.	
Configuration reference	CSI2_6412	
Initial conditions		
The SI2_112_command_01 is generated by using the SI2_112_command configuration file. The SI2_112_response_01 is verified by using the SI2_112_response configuration file. The procedure PSI2_6421 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**
1	The LBA sends SI2_112_command_01 to the SPBM.	RQ1206_132
2	The SPBM sends SI2_112_response_01 to the LBA containing aSi2GetSpbmCertificateOk.	RQ1206_140 RQ1206_141 RQ1206_142 RQ1206_143 RQ1206_144 RQ1206_145 RQ1206_146 RQ1206_147 RQ1206_148 RQ1206_149 RQ1206_150 RQ1206_152 RQ1206_153 RQ1206_154

12.6.4.3.1.3 SI2_64313 - Si2.GetSpbmCertificate - unsupported family identifier

Test ID	SI2_64313	
Test objectives	To verify that when the LBA requests SPBM credentials for an unsupported family identifier using a Si2.GetSpbmCertificate command the SPBM sends a response which includes eNotSupportedFamilyId.	
Configuration reference	CSI2_6411	
Initial conditions		
The SI2_113_commands are generated by using the SI2_113_command configuration file. The SI2_113_responses are verified by using the SI2_113_response configuration file. The procedure PSI2_6421 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**
1	The LBA sends SI2_113_command_01 to the SPBM.	
2	The SPBM sends SI2_113_response_01 to the LBA containing aSi2GetSpbmCertificateError with the error cause eNotSupportedFamilyId.	RQ1206_148a
3	The LBA sends SI2_113_command_02 to the SPBM.	
4	The SPBM sends SI2_113_response_02 to the LBA containing aSi2GetSpbmCertificateError with the error cause eNotSupportedFamilyId.	RQ1206_148b

12.6.4.3.1.4 SI2_63314 - Si2.GetSpbmCertificate - no trusted public key ID supported by SPBM

Test ID	SI2_63314	
Test objectives	To verify that the SPBM returns an error, when none of the trusted public key identifiers sent in the aSspPkIdListForSpbmVerification is supported by the SPBM.	
Configuration reference	CSI2_6411	
Initial conditions		
The SI2_114_command_01 is generated by using the SI2_114_command configuration file. The SI2_114_response_01 is verified by using the SI2_114_response configuration file. The procedure PSI2_6421 shall be successfully executed.		
Test sequence		
Step	Description	Requirements
1	The LBA sends SI2_114_command_01 to the SPBM.	
2	The SPBM sends SI2_114_response_01 to the LBA containing aSi2GetSpbmCertificateError with the error cause eNotSupportedPkIdSpbmVerification.	RQ1206_151a

12.6.4.3.1.5 SI2_64315 - Si2.GetSpbmCertificate - no trusted public key ID for SPBL verification supported

Test ID	SI2_64315	
Test objectives	To verify that the SPBM returns an error, when none of the trusted public key identifiers received in the aSspPkIdListForSpblVerification is supported.	
Configuration reference	CSI2_6411	
Initial conditions		
The SI2_115_command_01 is generated by using the SI2_115_command configuration file. The SI2_115_response_01 is verified by using the SI2_115_response configuration file. The procedure PSI2_6421 shall be successfully executed.		
Test sequence		
Step	Description	Requirements
1	The LBA sends SI2_116_command_01 to the SPBM.	
2	The SPBM sends SI2_116_response_01 to the LBA containing aSi2GetSpbmCertificateError with the error cause eNotSupportedEncryptionAlgorithm.	RQ1206_151d

12.6.4.3.1.6 SI2_64316 - Si2.GetSpbmCertificate - no supported encryption algorithm

Test ID	SI2_64316	
Test objectives	To verify that the SPBM returns an error, when none of the algorithm identifiers received in the aCipherAlgorithmList is supported.	
Configuration reference	CSI2_6411	
Initial conditions		
The SI2_116_command_01 is generated by using the SI2_116_command configuration file. The SI2_116_response_01 is verified by using the SI2_116_response configuration file. The procedure PSI2_6421 shall be successfully executed.		
Test sequence		
Step	Description	Requirements
1	The LBA sends SI2_116_command_01 to the SPBM.	
2	The SPBM sends SI2_116_response_01 to the LBA containing aSi2GetSpbmCertificateError with the error cause eNotSupportedEncryptionAlgorithm.	RQ1206_151d

12.6.4.3.1.7 SI2_64317 - Si2.GetSpbmCertificate - no supported SKID for SPBM verification

Test ID	SI2_64317	
Test objectives	To verify that the SPBM returns the error code eNotSupportedPkIdSpbmVerification when aSspPkIdListForSpbmVerification is not supported.	
Configuration reference	CSI2_6411	
Initial conditions		
The SI2_117_command_01 is generated by using the SI2_117_command configuration file. The SI2_117_response_01 is verified by using the SI2_117_response configuration file. The procedure PSI2_6421 shall be successfully executed.		
Test sequence		
Step	Description	Requirements
1	The LBA sends SI2_117_command_01 to the SPBM.	
2	The SPBM sends SI2_117_response_01 to the LBA containing aSi2GetSpbmCertificateError with the error cause eNotSupportedPkIdSpbmVerification.	RQ1206_151c

12.6.4.3.1.8 SI2_64318 - Si2.GetSpbmCertificate - no supported SKID for SPBL verification

Test ID	SI2_64318	
Test objectives	To verify that the SPBM returns the error code eNotSupportedPkIdSpblVerification when aSspPkIdListForSpblVerification is not supported.	
Configuration reference	CSI2_6411	
Initial conditions		
The SI2_118_command_01 is generated by using the SI2_118_command configuration file. The SI2_118_response_01 is verified by using the SI2_118_response configuration file. The procedure PSI2_6421 shall be successfully executed.		
Test sequence		
Step	Description	Requirements
1	The LBA sends SI2_118_command_01 to the SPBM.	
2	The SPBM sends SI2_118_response_01 to the LBA containing aSi2GetSpbmCertificateError with the error cause eNotSupportedPkIdSpblVerification.	RQ1206_151c

12.6.4.3.1.9 SI2_64319 - Si2.GetSpbmCertificate - no selection of a family identifier

Test ID	SI2_64319	
Test objectives	To verify that the SPBM returns eSpblSelectOneFamilyId with an error cause indicating that at least one family identifier shall be selected by the Secondary Platform Bundle Loader.	
Configuration reference	CSI2_6411	
Initial conditions		
The SI2_119_command_01 is generated by using the SI2_119_command configuration file. The SI2_119_response_01 is verified by using the SI2_119_response configuration file. The procedure PSI2_6421 shall be successfully executed.		
Test sequence		
Step	Description	Requirements
1	The LBA sends SI2_119_command_01 to the SPBM.	
2	The SPBM sends SI2_119_response_01 to the LBA containing aSi2GetSpbmCertificateError with the error cause eSpblSelectOneFamilyId.	RQ1206_148c

12.6.4.3.1.10 SI2_643110 - Si2.GetSpbmCertificate - no selection of an OID

Test ID	SI2_643110	
Test objectives	To verify that the SPBM returns eSpblSelectOneOid with an error cause indicating that at least one custodian shall be selected by the SPBL.	
Configuration reference	CSI2_6411	
Initial conditions		
The SI2_1110_command_01 is generated by using the SI2_1110_command configuration file. The SI2_1110_response_01 is verified by using the SI2_1110_response configuration file. The procedure PSI2_6421 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**
1	The LBA sends SI2_1111_command_01 to the SPBM.	
2	The SPBM sends SI2_1111_response_01 to the LBA containing aSi2GetSpbmCertificateError with the error cause eSpblSelectOneOid.	RQ1206_150

12.6.4.3.2 Si2.GetBoundSpbImage command and response handling

12.6.4.3.2.1 SI2_64321 - Si2.GetBoundSpbImage - normal process

Test ID	SI2_64321	
Test objectives	To verify that the LBA requests the SPBL bound image from the SPBM by sending an Si2GetBoundSpbImage command.	
Configuration reference	CSI2_6411	
Initial conditions		
The SI2_121_command_01 is generated by using the SI2_121_command configuration file. The SI2_121_response_01 is verified by using the SI2_121_response configuration file. The procedure PSI2_6421 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**
1	The LBA sends SI2_121_command_01 to the SPBM.	
2	The SPBM sends SI2_121_response_01 to the LBA. The LBA (tester) shall verify that the response is well formatted.	RQ1206_155 RQ1206_156 RQ1206_157 RQ1206_158 RQ1206_159 RQ1206_160 RQ1206_161 RQ1206_162 RQ1206_164 RQ1206_165 RQ1206_166 RQ1206_167 RQ1206_168 RQ1206_169 RQ1206_170 RQ1206_171 RQ1206_172 RQ1206_173 RQ1206_174 RQ1206_175 RQ1206_176 RQ1206_177 RQ1206_178 RQ1206_179 RQ1206_180 RQ1206_182 RQ1206_183 RQ1206_184

12.6.4.3.2.2 SI2_64322 - Si2.GetBoundSpbImage - no or invalid SSP credentials

Test ID	SI2_64322	
Test objectives	To verify that the SPBM returns aSi2GetBoundSpbImageError with the error code eInvalidBoundSpbImage when the LBA requests the SPBL bound image from the SPBM by sending Si2GetBoundSpbImage command without SSP credentials or with invalid SSP credentials.	
Configuration reference	CSI2_6411	
Initial conditions		
The SI2_122_command_01 is generated by using the SI2_122_command configuration file. The SI2_122_response_01 is verified by using the SI2_122_response configuration file. The procedure PSI2_6421 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**
1	The LBA sends SI2_122_command_01 to the SPBM (without or with invalid, SSP credentials).	
2	The SPBM sends SI2_122_response_01 response to the LBA. The LBA (tester) shall verify that the response contains aSi2GetBoundSpbImageError with the error code eInvalidSpbImage.	RQ1206_170 RQ1206_184e

12.6.4.3.2.3 SI2_64323 - Si2.GetBoundSpbImage - invalid aCodeM

Test ID	SI2_64323	
Test objectives	To verify that the SPBM returns aSi2GetBoundSpbImageError with the error code eInvalidCodeM when the LBA requests the SPBL bound image from the SPBM by sending Si2GetBoundSpbImage command with an unknown aCodeM included in the SSP credentials.	
Configuration reference	CSI2_6411	
Initial conditions		
The SI2_123_command_01 is generated by using the SI2_123_command configuration file. The SI2_123_response_01 is verified by using the SI2_123_response configuration file. The procedure PSI2_6421 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**
1	The LBA sends SI2_123_command_01 to the SPBM (with an unknown aCodeM).	
2	The SPBM sends SI2_123_response_01 response to the LBA. The LBA (tester) shall verify that the response is well formatted.	RQ1206_184

12.6.4.3.2.4 SI2_64324 - Si2.GetBoundSpbImage - invalid SPBL certificates

Test ID	SI2_64324	
Test objectives	To verify that the SPBM returns aSi2GetBoundSpbImageError with error code eInvalidSpblCertificate when the LBA requests the SPBL bound image from the SPBM by sending Si2GetBoundSpbImage command with an invalid certification path to the SPBL certificate.	
Configuration reference	CSI2_6411	
Initial conditions		
The SI2_124_command_01 is generated by using the SI2_124_command configuration file. The SI2_124_response_01 is verified by using the SI2_124_response configuration file. The procedure PSI2_6421 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**
1	The LBA sends SI2_124_command_01 to the SPBM (with an invalid certification path).	
2	The SPBM sends SI2_124_response_01 response to the LBA. The LBA (tester) shall verify that the response contains aSi2GetBoundSpbImageError with error code eInvalidSpblCertificate.	RQ1206_184a

12.6.4.3.2.5 SI2_64325 - Si2.GetBoundSpbImage - invalid ChallengeS

Test ID	SI2_64325	
Test objectives	To verify that the SPBM returns a response containing aSi2GetBoundSpbImageError with error code eInvalidChallengeS when the LBA requests the SPBL bound image from the SPBM and the ChallengeS returned by the SPBL does not match the one generated by the SPBM.	
Configuration reference	CSI2_6411	
Initial conditions		
The SI2_125_command_01 is generated by using the SI2_125_command configuration file. The SI2_125_response_01 is verified by using the SI2_125_response configuration file. The procedure PSI2_6421 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**
1	The LBA sends SI2_125_command_01 to the SPBM (with an invalid ChallengeS).	
2	The SPBM sends SI2_125_response_01 response to the LBA. The LBA (tester) shall verify that the response contains aSi2GetBoundSpbImageError with error code eInvalidChallengeS.	RQ1206_184c

12.6.4.3.2.6 SI2_64326 - Si2.GetBoundSpbImage - invalid selected SPB Image

Test ID	SI2_64326	
Test objectives	To verify that the SPBM is checking the validity of the SPB for the iSSP.	
Configuration reference	CSI2_6411	
Initial conditions		
The SI2_126_command_01 is generated by using the SI2_126_command configuration file. The SI2_126_response_01 is verified by using the SI2_126_response configuration file. The procedure PSI2_6421 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**
1	The LBA sends SI2_126_command_01 to the SPBM.	
2	The SPBM sends SI2_126_response_01 response to the LBA. The LBA (tester) shall verify that the response contains aSi2GetBoundSpbImageError with error code eInvalidSpbImage.	RQ1206_184d

12.6.4.3.2.7 SI2_64327 - Si2.GetBoundSpbImage - invalid TransacId

Test ID	SI2_64327	
Test objectives	To verify that the SPBM returns a response containing aSi2GetBoundSpbImageError with error code eInvalidBoundSpbImageByTransacId when the LBA requests the SPBL bound image from the SPBM by sending Si2GetBoundSpbImage command with an invalid referenced TransacId.	
Configuration reference	CSI2_6411	
Initial conditions		
The SI2_127_command_01 is generated by using the SI2_127_command configuration file. The SI2_127_response_01 is verified by using the SI2_127_response configuration file. The procedure PSI2_6421 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**
1	The LBA sends SI2_127_command_01 to the SPBM (with an invalid referenced TransacId.	
2	The SPBM sends SI2_127_response_01 response to the LBA. The LBA (tester) shall verify that the response contains aSi2GetBoundSpbImageError with error code eInvalidBoundSpbImageByTransacId.	RQ1206_184e

12.6.4.3.3 Si2.HandleNotification command and response handling

12.6.4.3.3.1 SI2_64331 - Si2.HandleNotificationCommand - normal process

Test ID	SI2_64333	
Test objectives	To verify that the "Si2HandleNotification" function is used by the LBA to send any notification about the result of the Secondary Platform Bundle management to the SPBM.	
Configuration reference	CSI2_6411	
Initial conditions		
The SI2_131_command_01 is generated by using the SI2_131_command configuration file. The SI2_131_response_01 is verified by using the SI2_131_response configuration file. The procedure PSI2_6421 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**
1	The LBA sends SI2_131_command_01 to the SPBM.	
2	The SPBM sends SI2_131_response_01 response to the LBA. The LBA (tester) shall verify that the response is well formatted.	RQ1206_185 RQ1206_186 RQ1206_187 RQ1206_188

NOTE: The response from the SPBM does not contain error codes. Invalid parameters within the Si2.HandleNotification command will not be indicated and nor can they be identified.

12.6.4.4 Requirements not testable, implicitly verified or verified elsewhere

The following requirements 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:

RQ1206_001, RQ1206_072, RQ1206_073, RQ1206_074, RQ1206_075, RQ1206_076, RQ1206_077, RQ1206_078, RQ1206_079, RQ1206_081 and RQ1206_096.

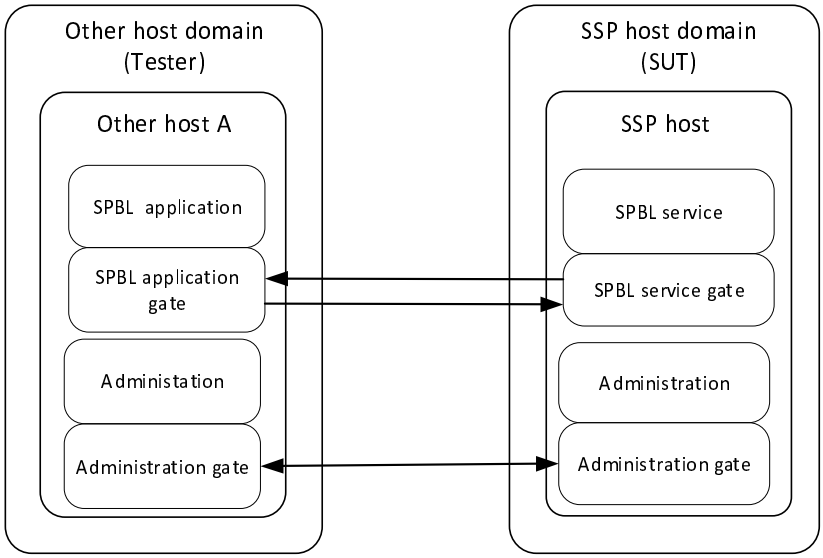
The following requirements are generated from descriptive text. A verification by tests defined within the present document is not possible:

RQ1206_083, RQ1206_098, RQ1206_104, RQ1206_112, RQ1206_122.

12.6.5 Si3 interface

12.6.5.1 Configurations

12.6.5.1.1 CSI3_6511 - SPBL service - host A

Configuration ID	CSI3_6511
Configuration description	

12.6.5.1.2 SSP configuration

The SSP under test shall be configured by the SSP Manufacturer (SSPM).

12.6.5.2 Procedures

12.6.5.2.1 PSI3_6521 - Pipe session opening on the SPBL service gate

Procedure ID	PSI3_6521
Objectives	The other host A shall be able to open a pipe session to the SPBL service gate of the SSP host. SPBL service identifier is defined as the OFL service identifier in Global Platform OFL VNP Extension [16].
Configuration reference	CSI3_6511
Initial conditions	
Test 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 SPBL service gate. • $GATE_{SPBL}$: The UUID gate identifier of the SPBL gate (BB780E30-419A-5B71-9B98-18A042E75899).
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 SPBL application gate. • $GATE_{SPBL}$: The UUID gate identifier of the SPBL gate (BB780E30-419A-5B71-9B98-18A042E75899).

12.6.5.2.2 PSI3_6522 - Verification of the SPBL service availability

Procedure ID	PSI3_6522
Objectives	The identity application gate shall verify the availability of the SPBL service gate identifier in the register GATE_LIST of the registry in the identity service gate.
Configuration reference	CSVC_311, CSI3_6511
Initial conditions	
The procedure PSI3_6521 shall execute successfully.	
Test sequence	
Step	**Description**
1	Identity application gate sends ANY_GET_PARAMETER command (pipe $PIPE_{XY}$) to the identity service gate with the register '04'H.
2	Identity service gate sends ANY_GET_PARAMETER response to the identity application gate. The service identifier 'BB780E30-419A-5B71-9B98-18A042E75899F917' shall be present.

12.6.5.3 Test descriptions

12.6.5.3.1 Si3.GetSspInfo command and response handling

12.6.5.3.1.1 SI3_65311 - Si3.GetSspInfo command with SpbFamilyId and an OID for the custodian, SSP with configuration for aSpbFamilyId and aCustodianOid

Test ID	SI3_65311	
Test objectives	To verify that the SPBL application gate (Other Host) is able to retrieve the correct SSP Information by sending a Si3.GetSspInfo command with SpdFamilyId and an OID for the custodian of the family identifier to the SPBL service gate if the SSP has a configuration for aSpbFamilyId and aCustodianOid.	
Configuration reference	CSI3_6511	
Initial conditions		
The procedure PSI3_6521 shall be successfully executed where the SSP has a configuration for aSpbFamilyId and aCustodianOid.		
Test sequence		
Step	**Description**	**Requirements**
1	SPBL application gate (Other Host) sends GET_SSP_INFO command with SpbFamilyId and Custodian Oid to the SPBL service gate.	RQ1206_190 RQ1206_191 RQ1206_195 RQ1206_274
2	SPBL service gate sends ANY_OK response with SspInfoPublic to the SPBL application gate. The response data is structured as follows: • aSpblSpecVerInfo shall be present with corresponding value defined for this version of specification. • aSspFamilyCryptoInfoBlock shall be present aSpbFamilyId shall be present a single aSspFamilyCryptoInfo shall be present aCustodianOid shall be present aSspOidCryptoInfo shall be present a list of trusted public key identifiers and a list of algorithm identifiers which can be used with that aSpbFamilyId and that aCustodianOid shall be present.	RQ1206_192 RQ1206_194 RQ1206_197 RQ1206_200 RQ1203_019 RQ1206_189 RQ1206_193 RQ1206_196

12.6.5.3.1.2 SI3_65312 - Si3.GetSspInfo command with SpbFamilyId only, SSP has configuration for SpbFamilyId

Test ID	SI3_65312	
Test objectives	To verify that the SPBL application gate (Other Host) is able to retrieve SSP Information by sending a Si3.GetSspInfo (GET_SSP_INFO) command with SpdFamilyId only to the SPBL service gate.	
Configuration reference	CSI3_6511	
Initial conditions		
The procedure PSI3_6521 shall be successfully executed where the SSP has a configuration for SpbFamilyId.		
Test sequence		
Step	**Description**	**Requirements**
1	SPBL application gate (Other Host) sends GET_SSP_INFO command with SpbFamilyId to the SPBL service gate.	
2	SPBL service gate sends ANY_OK response with SspInfoPublic to the SPBL application gate. • Verify that aSpblSpecVerInfo is present and is set to the release corresponding to the version of the implemented specification. • Verify that aSspFamilyCryptoInfoBlock is present Verify that aSpbFamilyId is present Verify that aSspOidCryptoInfoBlock is present for each supported custodian Verify that aCustodianOid is present Verify that aSspOidCryptoInfo is present Verify that the list of trusted public key identifiers and the list of algorithm identifiers which are to be used with that aSpbFamilyId and that aCustodianOid are present.	RQ1206_197 RQ1206_200

12.6.5.3.1.3 SI3_65313 - Si3.GetSspInfo command with SpbFamilyId, SSP has no configuration for SpbFamilyId

Test ID	SI3_65313	
Test objectives	To verify that the SPBL application gate (Other Host) is able to retrieve SSP Information by sending a Si3.GetSspInfo (GET_SSP_INFO) command with spdFamilyId only to the SPBL service gate.	
Configuration reference	CSI3_6511	
Initial conditions		
The procedure PSI3_6521 shall be successfully executed where the SSP has no configuration for SpbFamilyId.		
Test sequence		
Step	**Description**	**Requirements**
1	SPBL application gate (Other Host) sends GET_SSP_INFO command with SpbFamilyId to the SPBL service gate.	
2	SPBL service gate sends ANY_OK response with SspInfoPublic to the SPBL application gate: • Verify that aSpblSpecVerInfo is present and is set to the release corresponding to the version of the implemented specification. • Verify that aSspGeneralCryptoInfo is present Verify that a list of trusted public key identifiers and a list of algorithm identifiers are present, which are not associated with any family identifier and any custodian.	RQ1206_198 RQ1206_200

12.6.5.3.1.4 SI3_65314 - Si3.GetSspInfo command with empty parameters

Test ID	SI3_65314
Test objectives	To verify that the SPBL application gate (Other Host) is able to retrieve SSP Information by sending a Si3.GetSspInfo (GET_SSP_INFO) command with empty parameters to the SPBL service gate.
Configuration reference	CSI3_6511

Initial conditions		
The procedure PSI3_6521 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**
1	SPBL application gate (Other Host) sends GET_SSP_INFO command without SpbFamilyId and OID for the associated custodian to the SPBL service gate.	
2	SPBL service gate sends ANY_OK response with SspInfoPublic to the SPBL application gate: • Verify that aSpblSpecVerInfo is present and is set to the release corresponding to the version of the implemented specification. • Verify that aSspGeneralCryptoInfo is present for each supported aSpbFamilyId Verify that aSpbFamilyId is present Verify that aSpbOidCryptoInfoBlock is present Verify that a list of trusted public key identifiers and a list of algorithm identifiers are present, which are to be used with that aSpbFamilyId and the associated aCustodianOid.	RQ1206_199 RQ1206_200

12.6.5.3.2 Si3.SetSpbmCredential command and response handling

12.6.5.3.2.1 SI3_65321 - Si3.SetSpbmCredential

Test ID	SI3_65321	
Test objectives	To verify that the SPBL application gate (Other Host) is able to set SPBM Credentials by sending a Si3.SetSpbmCredential (ANY_SET_PARAMETER) to the SPBL Registry in the SPBL service gate.	
Configuration reference	CSI3_6511	
Initial conditions		
The procedure PSI3_6521 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**
1	SPBL application gate (Other Host) sends ANY_SET_PARAMETER command with index of IDS_CREDENTIAL_PARAMETER to the SPBL service gate.	RQ1206_201 RQ1206_202 RQ1206_204 RQ1203_028
2	SPBL service gate sends ANY_OK response to the SPBL application gate.	RQ1206_212 RQ1203_030 RQ1206_205 RQ1206_206 RQ1206_207

12.6.5.3.3 Si3.LoadBoundSpbInfo command and response handling

12.6.5.3.3.1 SI3_65331 - Si3.LoadBoundSpbInfo

Test ID	SI3_65331	
Test objectives	To verify that the SPBL application gate (Other Host) is able to load bound SPB Information by sending a Si3.LoadBoundSpbInfo (OFL_DO_OPERATE) command to the SPBL service gate.	
Configuration reference	CSI3_6511	
Initial conditions		
The procedure PSI3_6521 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**
1	SPBL application gate (Other Host) sends OFL_DO_OPERATE command with parameter "DoOperateParameter" to the SPBL service gate to Secondary Platform Bundle.	RQ1206_214 RQ1206_215
2	SPBL service gate sends ANY_OK response to the SPBL application gate.	RQ1206_229 RQ1203_040

12.6.5.3.4 Si3.LoadBoundSpbSds command and response handling

12.6.5.3.4.1 SI3_65341 - Si3.LoadBoundSpbSds

Test ID	SI3_65341	
Test objectives	To verify that the SPBL application gate (Other Host) is able to load bound SPB SDS by sending a Si3.LoadBoundSpbSds (OFL_CHANGE_SEGMENT) command to the SPBL service gate.	
Configuration reference	CSI3_6511	
Initial conditions		
The test SI3_65331 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**
1	SPBL application gate (Other Host) sends OFL_CHANGE_SEGMENT command with parameter "ChangeSegmentParameter" to the SPBL service gate to Secondary Platform Bundle.	RQ1206_230 RQ1206_232 RQ1206_233
2	SPBL service gate sends ANY_OK response to the SPBL application gate.	RQ1206_234

12.6.5.3.5 Si3.LoadBoundSpbSeg command and response handling

12.6.5.3.5.1 SI3_65351 - Si3.LoadBoundSpbSeg

Test ID	SI3_65351	
Test objectives	To verify that the SPBL application gate (Other Host) is able to load bound SPB Segment by sending a Si3.LoadBoundSpbSeg (OFL_LOAD_SEGMENT) command to the SPBL service gate.	
Configuration reference	CSI3_6511	
Initial conditions		
The test SI3_65341 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**
1	SPBL application gate (Other Host) sends OFL_LOAD_SEGMENT command with parameter "LoadSegmentParameter" to the SPBL service gate to Secondary Platform Bundle.	RQ1206_236 RQ1206_237 RQ1206_238
2	SPBL service gate sends ANY_OK response to the SPBL application gate.	RQ1206_239

12.6.5.3.6 Si3.GetSspCredential command and response handling

12.6.5.3.6.1 SI3_65361 - Si3.GetSspCredential

Test ID	SI3_65361	
Test objectives	To verify that the SPBL application gate (Other Host) is able to get SSP Credentials by sending a SI3.GetSspCredential (ANY_GET_PARAMETER) to the SPBL Registry in the SPBL service gate.	
Configuration reference	CSI3_6511	
Initial conditions		
The test SI3_65321 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**
1	SPBL application gate (Other Host) sends ANY_GET_PARAMETER command with index of TRE_CREDENTIAL_PARAMETER to the SPBL service gate.	RQ1206_241 RQ1206_242 RQ1206_243 RQ1203_031
2	SPBL service gate sends ANY_OK response to the SPBL application gate with value of TRE_CREDENTIAL_PARAMETER registry which contains SspCredential as defined in clause12.6.2.4 SSP credential of ETSI TS 103 666-2 [10].	RQ1206_244 RQ1203_032 RQ1206_203

12.6.5.3.7 Si3.EnableSpb command and response handling

12.6.5.3.7.1 SI3_65371 - Si3.EnableSpb

Test ID	SI3_65371	
Test objectives	To verify that the SPBL application gate is able to enable a SPB by sending a Si3.EnableSpb (OFL_ENABLE_FIRMWARE) to the SPBL service gate.	
Configuration reference	CSI3_6511	
Initial conditions		
The procedure PSI3_6521 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**
1	SPBL application gate sends OFL_ENABLE_FIRMWARE command to the SPBL service gate with the identifier of the Secondary Platform Bundle to enable.	RQ1206_246 RQ1206_247 RQ1204_003
2	SPBL service gate sends ANY_OK response to the SPBL application gate.	RQ1204_005

12.6.5.3.7.2 SI3_65372 - Si3.EnableSpb based on TELECOM_CAPABILITY value

Test ID	SI3_65372	
Test objectives	To verify that the SPBL application gate (Other Host) is able to enable no. of SPBs as defined in registry TELECOM_CAPABILITY.by sending a Si3.EnableSpb commands to the SPBL service gate.	
Configuration reference	CSI3_6511	
Initial conditions		
The procedure PSI3_6521 shall be successfully executed. Registry "TELECOM_CAPABILITY" is present in OFL Service Gate with at-least value 1.		
Test sequence		
Step	**Description**	**Requirements**
1	SPBL application gate (Other Host) sends ANY_GET_PARAMETER command with TELECOM_CAPABILITY registry index '80' to the SPBL service gate in Secondary Platform Bundle.	
2	SPBL service gate sends ANY_OK response to the SPBL application gate and provides value stored for TELECOM_CAPABILITY.	
3	The test SI3_65316 shall execute successfully - Si3.GetSspInfo.	
4	The test SI3_65321 shall execute successfully - Si3.SetSpbmCredential.	
5	The test SI3_65361 shall execute successfully - Si3.GetSspCredential.	
6	Execute step 7, step 8 and step 9 successfully, for count of TELECOM_CAPABILITY received in step 2.	
7	The test SI3_65331 shall execute successfully - Si3.LoadBoundSspInfo.	
8	Load Telecom SPB Image. Loop (until whole SPB Image is loaded): • The test SI3_65341 shall execute successfully - Si3.LoadBoundSpbSds. • The test SI3_65351 shall execute successfully - Si3. LoadBoundSpbSeg.	
9	The test SI3_65371 shall execute successfully - Si3.EnableSpb: • With SpbId of Telecom SPB Image.	
10	Execute step 7 and step 8 successfully: Execute step 9 with failure [eSPBL_E_EXCEED_TELECOM_CAPABILITY – '15'].	RQ1206_249 RQ1204_004

12.6.5.3.8 Si3.DisableSpb command and response handling

12.6.5.3.8.1 SI3_65381 - Si3.DisableSpb

Test ID	SI3_65381
Test objectives	To verify that the SPBL application gate is able to disable a SPB by sending a Si3.DisableSpb (OFL_DISABLE_FIRMWARE) to the SPBL service gate.
Configuration reference	CSI3_6511

Initial conditions		
The procedure PSI3_6521 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**
1	SPBL application gate sends OFL_DISABLE_FIRMWARE command to the SPBL service gate with the identifier of the Secondary Platform Bundle to disable.	RQ1206_252 RQ1206_253
2	SPBL service gate sends ANY_OK response to the SPBL application gate.	RQ1204_009 RQ1204_013

12.6.5.3.9 Si3.DeleteSpb command and response handling

12.6.5.3.9.1 SI3_65391 - Si3.DeleteSpb

Test ID	SI3_65391	
Test objectives	To verify that the SPBL application gate is able to delete a SPB by sending a Si3.DeleteSpb (OFL_DELETE_SESSION) to the SPBL service gate.	
Configuration reference	CSI3_6511	
Initial conditions		
The procedure PSI3_6521 shall execute successfully.		
Test sequence		
Step	**Description**	**Requirements**
1	SPBL application gate sends OFL_DELETE_SESSION command to the SPBL service gate with the identifier of the Secondary Platform Bundle to delete.	RQ1206_257 RQ1204_014
2	SPBL service gate sends ANY_OK response to the SPBL application gate.	RQ1204_015

12.6.5.3.10 Si3.GetSpbMetadata command and response handling

12.6.5.3.10.1 SI3_653101 - Si3.GetSpbMetadata

Test ID	SI3_653101	
Test objectives	To verify that the SPBL application gate (Other Host) shall be able to retrieve SPB metadata by sending a Si3.GetSpbMetadata (GET_SPB_METADATA) command to the SPBL service gate.	
Configuration reference	CSI3_6511	
Initial conditions		
The procedure PSI3_6521 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**
1	SPBL application gate (Other Host) sends GET_SPB_METADATA command with SpbId to the SPBL service gate to Secondary Platform Bundle.	RQ1206_190 RQ1206_191 RQ1206_260 RQ1206_261 RQ1206_262 RQ1206_054
2	SPBL service gate sends ANY_OK response to the SPBL application gate.	RQ1206_192 RQ1206_264 RQ1204_017 RQ1204_018

12.6.5.3.11 Si3.UpdateSpbState command and response handling

12.6.5.3.11.1 SI3_653111 - Si3.UpdateSpbState

Test ID	SI3_653111
Test objectives	To verify that the SPBL application gate (Other Host) is able to update SpbId by sending a Si3.UpdateSpbState (ANY_SET_PARAMETER) command to the SPBL service gate.
Configuration reference	CSI3_6511

Initial conditions		
The procedure PSI3_6521 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**
1	SPBL application gate (Other Host) sends ANY_SET_PARAMETER command with SpbId and SPB_ID registry index to the SPBL service gate in Secondary Platform Bundle.	RQ1206_266 RQ1206_267 RQ1206_268 RQ1204_019
2	SPBL service gate sends ANY_OK response to the SPBL application gate.	RQ1206_269

12.6.5.3.12 Si3.GetSpbState command and response handling

12.6.5.3.12.1 SI3_653121 - Si3.GetSpbState

Test ID	SI3_653121
Test objectives	To verify that the SPBL application gate (Other Host) is able to get the SPB state by sending a Si3.GetSpbState (ANY_SET_PARAMETER) command to the SPBL service gate.
Configuration reference	CSI3_6511

Initial conditions		
The procedure PSI3_6521 shall be successfully executed. The test SI3_653111 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**
1	SPBL application gate (Other Host) sends ANY_SET_PARAMETER command with SPB_STATE registry index to the SPBL service gate in Secondary Platform Bundle.	RQ1206_271 RQ1206_272 RQ1204_022
2	SPBL service gate sends ANY_OK response to the SPBL application gate.	RQ1206_273 RQ1204_023

12.6.5.3.13 SI3_65313 - SPB Management Operations

Test ID	SI3_65313
Test objectives	To verify that the SPBL application gate (Other Host) is able to load SPB and perform SPB management operations by sending Si3 layer commands to the SPBL service gate.
Configuration reference	CSI3_6511

Initial conditions		
The procedure PSI3_6521 shall be successfully executed.		
Test sequence		
Step	**Description**	**Requirements**
1	The test SI3_65316 shall be successfully executed - Si3.GetSspInfo.	RQ1203_017
2	The test SI3_65321 shall be successfully executed - Si3.SetSpbmCredential.	
3	The test SI3_65361 shall be successfully executed - Si3.GetSspCredential.	RQ1206_211 RQ1206_240
4	The test SI3_65331 shall be successfully executed - Si3.LoadBoundSspInfo For Non-Telecom SPB Image.	
5	Load Non-Telecom SPB Image. Loop (until whole SPB Image is loaded): • The test SI3_65341 shall be successfully executed - Si3.LoadBoundSpbSds The test SI3_65351 shall be successfully executed - Si3. LoadBoundSpbSeg.	RQ1206_230 RQ1206_235
6	The test SI3_65331 shall be successfully executed - Si3.LoadBoundSspInfo For Telecom SPB Image – 1.	
7	Load Telecom SPB Image - 1. Loop (until whole SPB Image is loaded): • The test SI3_65341 shall be successfully executed - Si3.LoadBoundSpbSds The test SI3_65351 shall be successfully executed y - Si3. LoadBoundSpbSeg.	
8	The test SI3_65331 shall execute successfully - Si3.LoadBoundSspInfo For Telecom SPB Image – 2.	

9	Load Telecom SPB Image - 2. Loop (until whole SPB Image is loaded): • The test SI3_65341 shall execute successfully - Si3.LoadBoundSpbSds The test SI3_65351 shall execute successfully - Si3. LoadBoundSpbSeg.	
10	The test SI3_65310 shall execute successfully - Si3.GetSpbMetadata With SpbId of Non-Telecom SPB Image.	
11	The test SI3_65371 shall execute successfully - Si3.EnableSpb With SpbId of Non-Telecom SPB Image.	RQ1206_245
12	The test SI3_65312 shall execute successfully - Si3.UpdateSpbState With SpbId of Non-Telecom SPB Image.	RQ1206_265
13	The test SI3_65313 shall execute successfully - Si3.GetSpbState • With SpbId of Non-Telecom SPB Image Verify returned state is Enable.	
14	The test SI3_65381 shall execute successfully - Si3.DisableSpb With SpbId of Non-Telecom SPB Image.	RQ1206_251
15	The test SI3_65312 shall execute successfully - Si3.UpdateSpbState With SpbId of Non-Telecom SPB Image.	
16	The test SI3_65313 shall execute successfully - Si3.GetSpbState • With SpbId of Non-Telecom SPB Image Verify returned state is Disable.	RQ1206_255 RQ1206_270
17	The test SI3_65391 shall execute successfully - Si3.DeleteSpb With SpbId of Non-Telecom SPB Image.	RQ1206_256

12.6.5.3.14 SI3_65314 - Si3.SwitchSpb

Test ID	SI3_65314	
Test objectives	To verify that the SPBL application gate (Other Host) shall be able to switch between different telecom SPBs by sending and perform SPB management operations by sending Si3 layer commands to the SPBL service gate.	
Configuration reference	CSI3_6511	
Initial conditions		
The procedure PSI3_6521 shall be successfully executed. The test SI3_65314 shall be successfully executed. Registry "TELECOM_CAPABILITY" is present in OFL Service Gate with at-least value 1.		
Test sequence		
Step	**Description**	**Requirements**
1	The test SI3_65371 shall be successfully executed - Si3.EnableSpb • With SpbId of Telecom SPB Image-1.	
2	The test SI3_65312 shall be successfully executed - Si3.UpdateSpbState With SpbId of Telecom SPB Image-1.	
3	The test SI3_65313 shall execute successfully - Si3.GetSpbState • With SpbId of Telecom SPB Image-1 Verify returned state is Enable.	
4	The test SI3_65312 shall execute successfully - Si3.UpdateSpbState • With SpbId of Telecom SPB Image-2.	
5	The test SI3_65313 shall execute successfully - Si3.GetSpbState • With SpbId of Telecom SPB Image-2 Verify returned state is Disable.	
6	SPBL application gate (Other Host) sends SWITCH_TELECOM_SPB command (Si3.SwitchSpb) with: • SpbId of Telecom SPB Image-1 as aSpbIdToBeDisabled; and • SpbId of Telecom SPB Image-2 as aSpbIdToBeEnabled to the SPBL service gate to Secondary Platform Bundle.	RQ1206_190 RQ1206_191
7	SPBL service gate sends ANY_OK response to the SPBL application gate.	RQ1206_192
8	The test SI3_65312 shall execute successfully - Si3.UpdateSpbState With SpbId of Telecom SPB Image-1.	
9	The test SI3_65313 shall execute successfully - Si3.GetSpbState • With SpbId of Telecom SPB Image-1 Verify returned state is Disable.	
10	The test SI3_65312 shall execute successfully - Si3.UpdateSpbState With SpbId of Telecom SPB Image-2.	
11	The test SI3_65313 shall execute successfully - Si3.GetSpbState • With SpbId of Telecom SPB Image-2 Verify returned state is Enable.	RQ1206_250

12.6.5.4 Requirements not testable, implicitly verified or verified elsewhere

The following requirements are generated from descriptive text. A verification by tests defined within the present document is not possible:

RQ1206_004, RQ1206_022, RQ1206_055, RQ1206_056, RQ1206_188, RQ1206_189, RQ1206_196, RQ1206_205, RQ1206_206, RQ1206_207, RQ1206_208, RQ1206_209, RQ1206_210, RQ1206_211, RQ1206_213, RQ1206_216, RQ1206_217, RQ1206_218, RQ1206_219, RQ1206_220, RQ1206_221, RQ1206_222, RQ1206_223, RQ1206_224, RQ1206_225, RQ1206_226, RQ1206_231, RQ1206_234, RQ1206_248, RQ1206_254, RQ1206_259, RQ1206_263.

12.6.6 Si4 interface

12.6.6.0 Si4 Principles

12.6.6.0.1 Si4 tunneling over Si3 and Si2

Figure 12.1 illustrates the tunneling of the Si4 over Si2 and Si3.

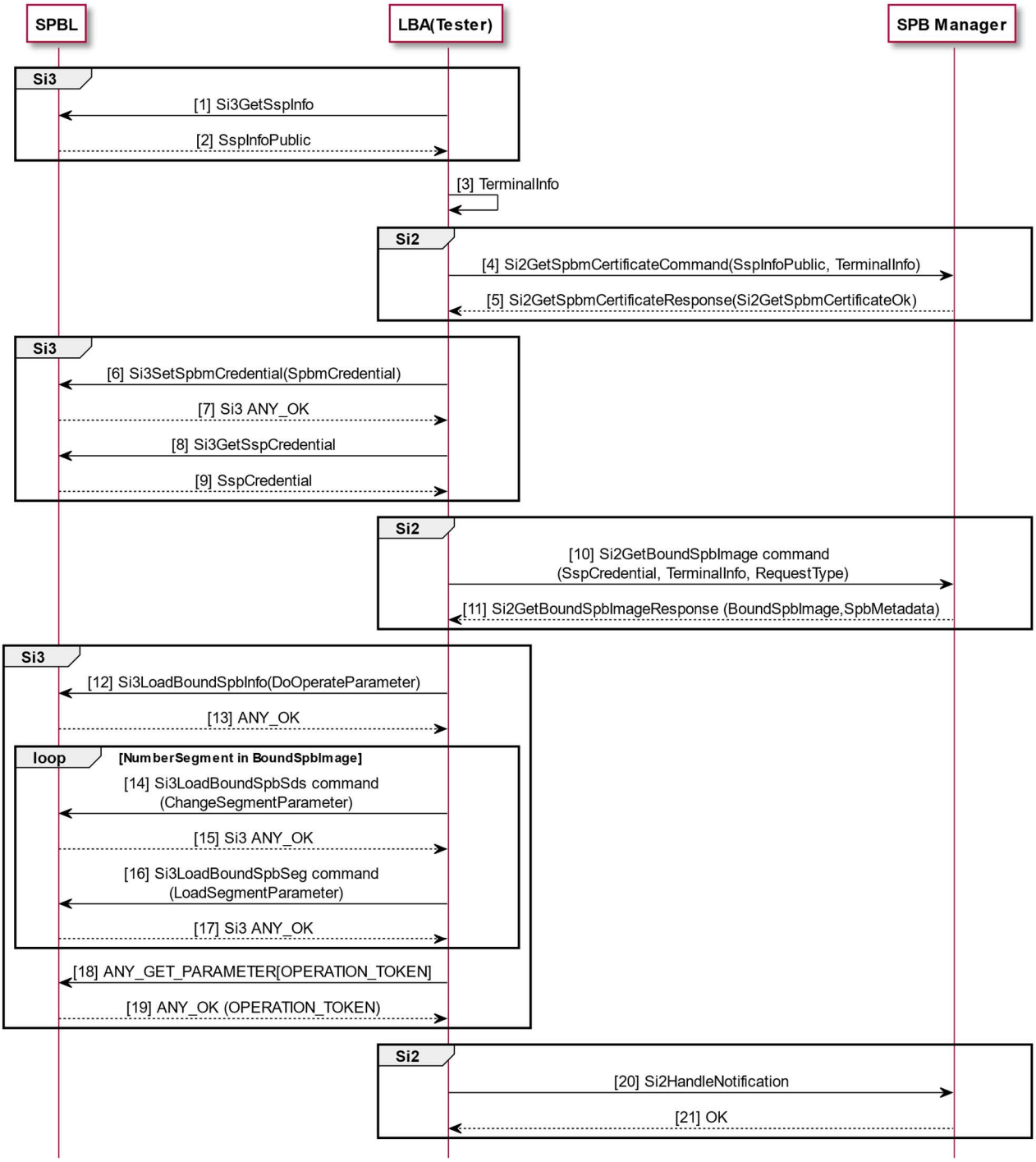


Figure 12.1: Security Protocol over Si2 and Si3

12.6.6.0.2 Si4 security protocol abstract view

Figure 12.2 illustrates the handling of the Si4 security protocol between SPBL and SPBM.

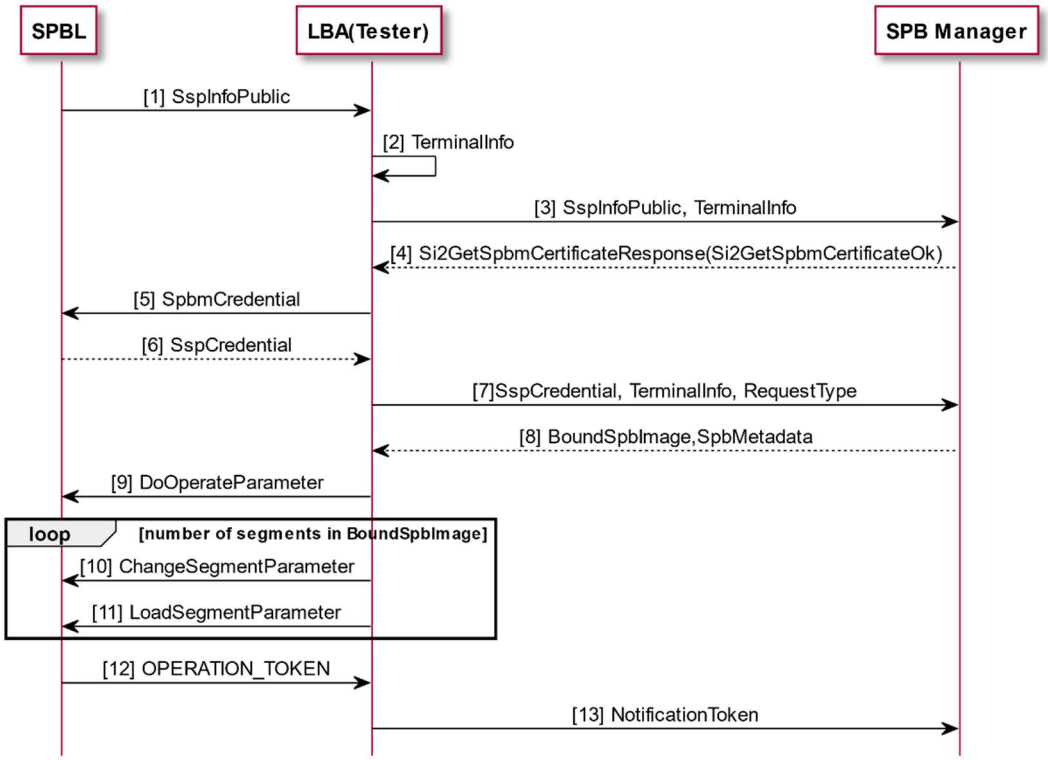


Figure 12.2: Si4 Security Protocol SPBL/SPB Manager

The Protocol Data Unit (PDU) conveying the Si4 security protocol are generated by using Si3 and Si2 messages. All PDUs are sequentially dependent and cannot be generated independently. In order to link these PDUs, a software tooling is available in the ETSI forge repository - SCP iSSP tooling [34].

12.6.6.0.3 Testing the Si4 SPBL service

Figure 12.3 illustrates the testing of the SPBL service (SUT) from the LBA (tester). The Tester emulates the SPBM.

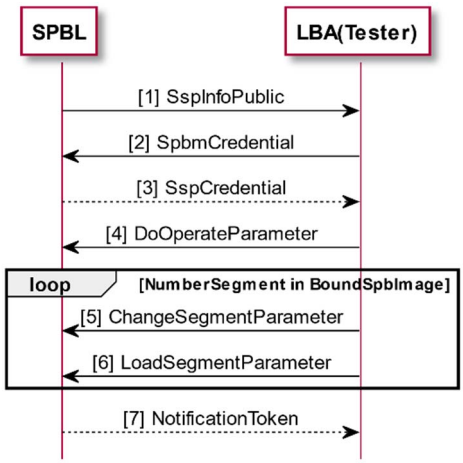


Figure 12.3: Si4 SPBL service

For sake of simplicity, the returned status subsequent to any exchange are not showed.

12.6.6.0.4 Testing the Si4 SPB Manager service

Figure 12.4 illustrates the testing of the SPBM service(SUT) from the LBA (tester). The Tester emulates the SPBL.

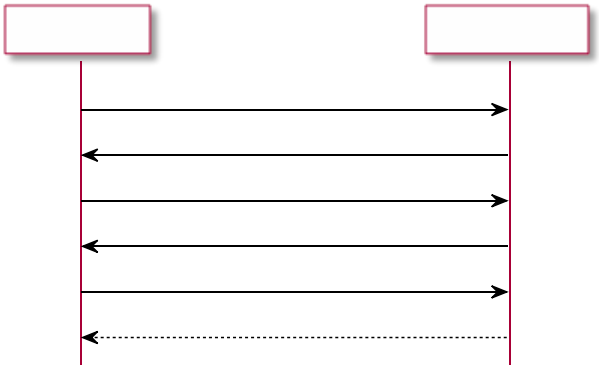
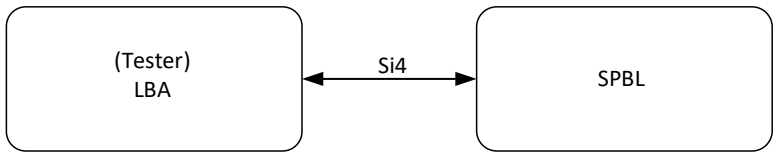


Figure 7: Si4 SPBM service

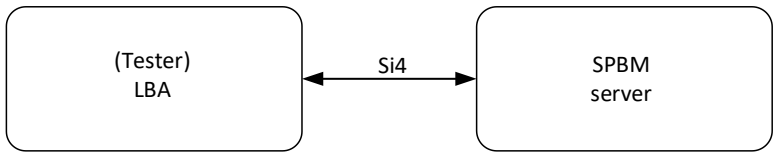
For sake of simplicity, the returned status subsequent to any exchange are not showed.

12.6.6.1 Configurations

12.6.6.1.1 CSI4_6611 - LBA - SPBL (SUT)

Configuration ID	CSI4_6611
Configuration description	

12.6.6.1.2 CSI4_6612 - LBA - SPB Manager (SUT)

Configuration ID	CSI4_Y12
Configuration description	

12.6.6.2 Procedures

12.6.6.2.1 PSI4_6621 - session opening between SPBL and the SPB Manager

Procedure ID	PSI4_6621
Procedure objectives	The LBA manages the bridge between the SPBL and the SPBM. The semantic of the Si4 security protocol is conveyed by using Si3 and Si2. The LBA is in charge to get and forward the Si4 data between the SPBL and the SPBM. For tests considerations, the LBA is transparent between SPBL and SPBM.
Configuration reference	CSI4_6611 or CSI4_6612
Initial conditions	
Test sequence	
Step	**Description**
1	The SPBL establishes the Si3 connection to the LBA.
2	The SBPM establishes the Si2 connection to the LBA.

12.6.6.3 Test Descriptions

12.6.6.3.1 Si4 - SPBL service

12.6.6.3.1.1 SI4_66311 - Normal flow

Test ID	SI4_66311	
Test objectives	To verify that no errors occur if the LBA (tester) stimulates the SPBL (SUT) with PDUs, and these PDUs are conveyed in Si4 semantic.	
Configuration reference	CSI4_6611	
Initial conditions		
The procedure PSI4_6621 shall be successfully executed. The SI4_111_pdu is generated by using the SI4_111_pdu configuration file.		
Test sequence		
Step	**Description**	**Requirements**
1	The SPBL sends to SI4_111_pdu_01 to the LBA.	RQ1206_248 RQ1206_254 RQ1206_258 RQ1206_259 RQ1206_263
2	The LBA sends to SI4_111_pdu_02 to the SPBL.	RQ1206_217 RQ1206_207 RQ1206_209 RQ1206_210 RQ1202_001 RQ1202_002 RQ1202_006 RQ1202_007 RQ1202_008 RQ1202_009 RQ1202_010

3	The SPBL sends to SI4_111_pdu_02 to the LBA.	RQ1206_237 RQ1202_006 RQ1202_007 RQ1202_008 RQ1202_009 RQ1202_010 RQ1202_011 RQ1202_012 RQ1202_013 RQ1202_014 RQ1202_015 RQ1202_016 RQ1202_017 RQ1202_030 RQ1202_033 RQ1202_042
4	The LBA sends to SI4_111_pdu_04 to the SPBL.	RQ1206_218 RQ1206_219 RQ1206_220 RQ1206_221 RQ1206_222 RQ1206_223 RQ1206_224 RQ1206_225 RQ1206_226 RQ1206_226a RQ1206_226b RQ1202_006 RQ1202_007 RQ1202_008 RQ1202_009 RQ1202_010
5	The LBA sends to SI4_111_pdu_05 to the SPBL.	RQ1202_042
6	The LBA sends to SI4_111_pdu_06 to the SPBL.	RQ1206_231 RQ1202_038 RQ1202_039 RQ1202_040 RQ1202_041 RQ1202_042
7	The SPBL sends to SI4_111_pdu_07 to the LBA.	RQ1206_062 RQ1206_063 RQ1206_064 RQ1206_065 RQ1206_066 RQ1206_067 RQ1206_068 RQ1206_069 RQ1206_070 RQ1206_071

12.6.6.3.2 Si4 - SPB Manager service

12.6.6.3.2.1 SI4_66321 - Normal flow

Test ID	SI4_66321	
Test objectives	To verify that no errors occur if the LBA (tester) stimulates the SPBM (SUT) with PDUs, and these PDUs are conveyed in Si4 semantic.	
Configuration reference	CSI4_6612	
Initial conditions		
The procedure PSI4_6621 shall be successfully executed. The SI4_121_pdu is generated by using the SI4_121_pdu configuration file.		
Test sequence		
Step	**Description**	**Requirements**
1	The LBA sends to SI4_121_pdu_01 to the SPBM.	RQ1203_026

2	The SPBM sends to SI4_121_pdu_02 to the LBA.	RQ1202_015 RQ1202_016 RQ1202_017 RQ1202_018 RQ1202_019 RQ1202_020 RQ1202_021 RQ1202_022 RQ1202_023 RQ1202_024 RQ1202_025 RQ1202_026 RQ1202_027 RQ1202_028
3	The LBA sends to SI4_121_pdu_02 to the SPBM.	RQ1202_001 RQ1202_002 RQ1202_003 RQ1202_004 RQ1202_006 RQ1202_007 RQ1202_008 RQ1202_009 RQ1202_010
4	The SPBM sends to SI4_121_pdu_04 to the LBA.	RQ1202_015 RQ1202_016 RQ1202_017 RQ1202_018 RQ1202_019 RQ1202_020 RQ1202_021 RQ1202_022 RQ1202_023 RQ1202_024 RQ1202_025 RQ1202_026 RQ1202_027 RQ1202_028 RQ1202_005 RQ1202_030 RQ1202_033 RQ1202_038 RQ1202_039 RQ1202_040 RQ1202_041

12.6.6.4 Requirements not testable, implicitly verified or verified elsewhere

The following requirements are generated from descriptive text. A verification by tests defined within the present document is not possible:

RQ1202_005, RQ1202_029, RQ1202_031, RQ1202_032, RQ1202_034, RQ1202_035, RQ1202_036, RQ1202_037.

Annex A (informative): Core specification version information

Unless otherwise specified, the versions of ETSI TS 103 666-2 [10] from which conformance requirements have been extracted are as follows:

Release	Latest version from which conformance requirements have been extracted
15	V15.3.0 (2020-09)

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**
08/11/2021	SCP#102	SCP(21)000160r2	-	-	-	Version 15.0.0 first publication	-	15.0.0

History

Document history		
V15.0.0	December 2021	Publication