ETSI TS 103 834-1 V17.0.0 (2022-12)

TECHNICAL SPECIFICATION

Smart Secure Platform (SSP);
Part 1: Technical Specification, SSP Test Tool Interface
(Release 17)

Reference

DTS/SET-00103834-1vh00

Keywords

SCL, SSP, Test Tool Interface

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

Important notice

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

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

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

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

If you find a security vulnerability in the present document, please report it through our Coordinated Vulnerability Disclosure Program:
https://www.etsi.org/standards/coordinated-vulnerability-disclosure

Notice of disclaimer & limitation of liability

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.
No recommendation as to products and services or vendors is made or should be implied.
No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.
In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

Copyright Notification

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.

BLUETOOTH® is a trademark registered and owned by Bluetooth SIG, Inc.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Secure Element Technologies (SET).

The present document is part 1 of a multi-part deliverable covering the Test Tool Interface (TTI) for the Smart Secure Platform (SSP), as identified below:

Part 1: "Technical Specification, SSP Test Tool Interface";

Part 2: "Test Specification, SSP Test Tool Interface".

The contents of the present document are subject to continuing work within TC SET and may change following formal TC SET approval. If TC SET 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 SET for information;

2 presented to TC SET for approval;

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

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

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

Modal verbs terminology

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

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

1 Scope

The present document is part of a series of documents specifying the Test Tool Interface (TTI) for the Smart Secure Platform. It is the technical specification for the Test Tool Interface (TTI) shown in the test environment of ETSI TS 103 999-1 [3].

It describes:

- the setup of the (TTI);
- the authentication of the TT accessor granting rights for accessing TTI resources;
- the TTI resources;
- the principle and requirements for testing the SSP Primary Platform (also known as VPP).

The TTI should fulfil the related requirements from ETSI an Global Platform, acting as an interface between SUT and Test Tool, capable to smoothly handle the data exchange, consisting of controls, events and SCL packets.

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

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

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

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

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

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

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

[4] ETSI TS 103 999-2: "Smart Secure Platform (SSP); Part 2: Integrated SSP (iSSP) characteristics, Test Specification".

[5] GlobalPlatform: "Virtual Primary Platform - Network Protocol", Version 2.0.

NOTE: Available at: https://globalplatform.org/specs-library/globalplatform-technology-virtual-primary-platform/.

[6] ETSI TS 103 834-2: "Smart Secure Platform (SSP); Part 2: Test Specification, SSP Test Tool Interface".

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

[8] OASIS Standard: "MQTT Version 5.0", 07 March 2019".

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

[10] IETF RFC 7159: "The JavaScript Object Notation (JSON) Data Interchange Format".

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

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

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

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

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

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

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

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

[19] Recommendation ITU-T X.680 (02/2021): "Information technology - Abstract Syntax Notation One (ASN.1): Specification of basic notation".

[20] Recommendation ITU-T X.690 (02/2021) and Erratum 1 (09/2021): "Information technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)".

[21] Recommendation ITU-T X.501 (10/2019) and Amendment 1 (10/2021): "Information technology - Open Systems Interconnection - The Directory: Models".

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

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

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

[i.1] ETSI TS 103 666-3: "Smart Secure Platform (SSP); Part 3: Embedded SSP (eSSP) Type 1 characteristics".

[i.2] ETSI TS 103 666-4: "Smart Secure Platform (SSP); Part 4: Embedded SSP (eSSP) Type 2 characteristics".

[i.3] ResearchGate: "Attacking the Baseband Modem of Mobile Phones to Breach the Users' Privacy and Network Security"; DOI: 10.1109/CYCON.2015.7158480.

[i.4] IETF RFC 7668: "IPv6 over BLUETOOTH® Low Energy".

[i.5] Microsoft®: "Overview of Remote NDIS (RNDIS)".

NOTE: Available at https://learn.microsoft.com/en-us/windows-hardware/drivers/network/overview-of-remote-ndis--rndis-.

[i.6] USB: "Communications Class Subclass Specification for Ethernet Emulation Model Devices".

NOTE: Available at https://www.usb.org/document-library/cdc-subclass-specification-ethernet-emulation-model-devices-10.

3 Definition of terms, symbols, abbreviations, forms and ASN.1 syntax

3.1 Terms

For the purposes of the present document, the terms given in ETSI TS 103 666-1 [1], ETSI TS 103 999-1 [3] and the following apply:

impersonated host: virtual host belonging to the SUT host network but executed from a host in the TTI host network

NOTE: This host is defined in clause 6.3.2.2 of the present document.

3.2 Symbols

For the purposes of the present document, the symbols given in ETSI TS 103 666-1 [1] and ETSI TS 103 999-1 [3] apply.

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in ETSI TS 103 666-1 [1], ETSI TS 103 999-1 [3], and the following apply:

AAA	Accessor Authentication Application
AAS	Accessor Authentication Service
OEM	Original Equipment Manufacturer (Terminal maker)
PP	Primary Platform
PPI	Primary Platform Interface
RDE	Router Data Extractor
TT	Test Tool
TTI	Test Tool Interface
TTI_UL	Test Tool Interface UnderLayer
TTM	Test Tool Maker
URL	Universal Resource Locator
USB	Universal Serial Bus

3.4 Formats

3.4.1 Numbers and Strings

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

Table 3.1: Convention of Numbering and Strings

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

3.5 ASN.1 syntax

3.5.1 Introduction

The description of some data objects in the present document is based on ASN.1 specified in Recommendation ITU-T X.680 [19] and encoded in TLV structures using DER (Distinguished Encoding Rule) encoding as specified in Recommendation ITU-T X.690 [20]. This provides a flexible description of those data objects. The complete ASN.1 code is divided into a number of ASN.1 sections in the specifications. In order to facilitate the extraction of the complete ASN.1 code from the specification, each ASN.1 section begins with a text paragraph consisting entirely of an ASN.1 start tag, which consists of a double hyphen followed by a single space and the text string "ASN1START" (in all upper case letters). Each ASN.1 section ends with a text paragraph consisting entirely of an ASN.1 stop tag, which consists of a double hyphen followed by a single space and the text "ASN1STOP" (in all upper case letters).

The complete ASN.1 code may be extracted by copying all the text paragraphs between an ASN.1 start tag and the following ASN.1 stop tag in the order they appear, throughout the present document.

3.5.2 Start of ASN.1

```
-- ASN1START

TTIDefinitions { itu-t (0) identified-organization (4) etsi (0) smart-secure-platform (3834) part1
(1) }
DEFINITIONS
AUTOMATIC TAGS
EXTENSIBILITY IMPLIED ::=
BEGIN
EXPORTS ALL;
/* Imports */
IMPORTS
   Version,
   AccessorRights,
   AccessControl,
   UUID
FROM SSPDefinitions;

-- ASN1STOP
```

NOTE: The ASN.1 code is ended in Annex F.

4 Requirements for the TTI

As the TTI shall be useable in test environments as defined in ETSI TS 103 999-1 [3], requirements from various specifications shall be considered. Where the TTI shall be capable to fulfil requirements identified in:

- ETSI TS 103 666-1 [1] Smart Secure Platform (SSP); Part 1: General characteristics:
  - A listing of requirements to the TTI derived from ETSI TS 103 666-1 [1], clause 6.13, related to accessor authentication is available in ETSI TS 103 834-2 [6], clause 4.1.1.
  - A listing of requirements to the TTI derived from ETSI TS 103 666-1 [1], clause 8.3, related to the protocol layers is available in ETSI TS 103 834-2 [6], clause 4.1.2.
- ETSI TS103 666-2 [2] Smart Secure Platform (SSP); Part 2: Integrated SSP (iSSP) characteristics:
  - A listing of requirements to the TTI derived from ETSI TS 103 666-2 [2], clause 6.6, related to the runtime model is available in ETSI TS 103 834-2 [6], clause 4.2.1.
- ETSI TS 103 999-1 [3] Smart Secure Platform (SSP); Part 1: Test Specification, general characteristics:
  - A listing of requirements to the TTI derived from ETSI TS 103 999-1 [3], clause 4.2.3, related to the TTI requirements is available in ETSI TS 103 834-2 [6], clause 4.3.1.
- GlobalPlatform, Technology, Virtual Primary Platform [5]:
  - A listing of requirements to the TTI derived from GlobalPlatform, Technology, Virtual Primary Platform [5], will be made available in ETSI TS 103 834-2 [6], clause 4.4 if applicable.
- ETSI TS103 834-1 Smart Secure Platform (SSP); Part 2: Test Specification, SSP Test Tool Interface (the present document):
  - A listing of requirements to the TTI derived from the present document clause 5.2.3, related to the network protocol stack is available in ETSI TS 103 834-2 [6], clause 4.5.1.
  - A listing of requirements to the TTI derived from the present document clause 6.2.1, related to the TTI control service is available in ETSI TS 103 834-2 [6], clause 4.5.2.
  - A listing of requirements to the TTI derived from the present document clause 6.3.1, related to the TTI data service is available in ETSI TS 103 834-2 [6], clause 4.5.3.

5 Test Tool Interface (TTI) architecture

5.1 TTI environment

The TTI is to be integrated into the test environment defined in ETSI TS 103 999-1 [3] (see figure 5.1).

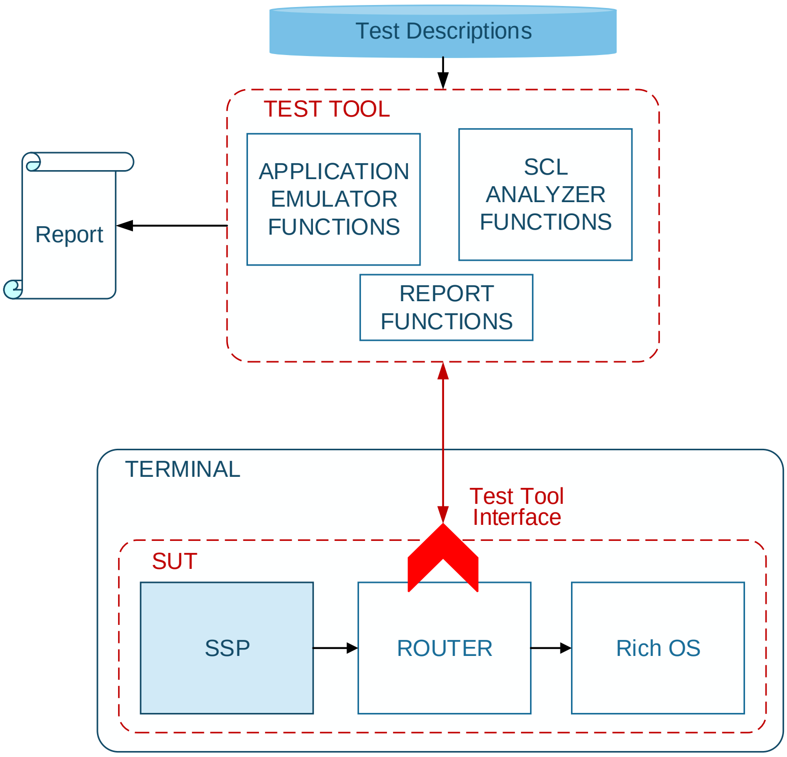

Figure 5.1: SSP test environment overview (according to ETSI TS 103 999-1 [3])

Figure 5.2 illustrates the data exchange between TT and TTI as defined in the ETSI TS 103 999-1 [3], clause 4.2.1.

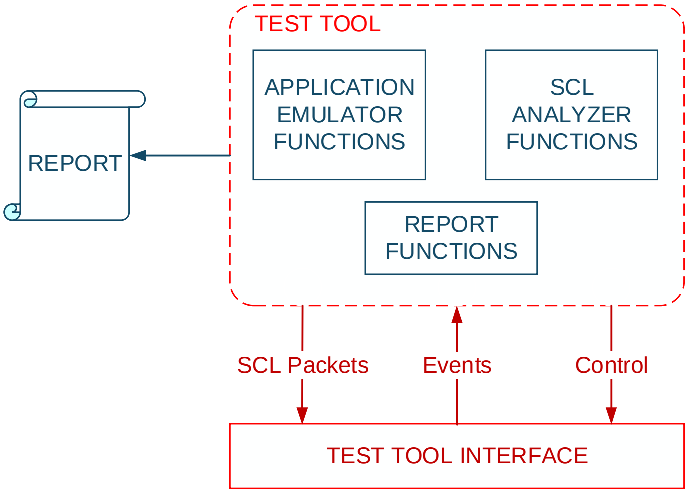

Figure 5.2: Data exchange between test tool and test tool interface

5.2 Network Protocol stack

5.2.1 TTI layers

Figure 5.3 illustrates the protocol stack defining the TTI layers.

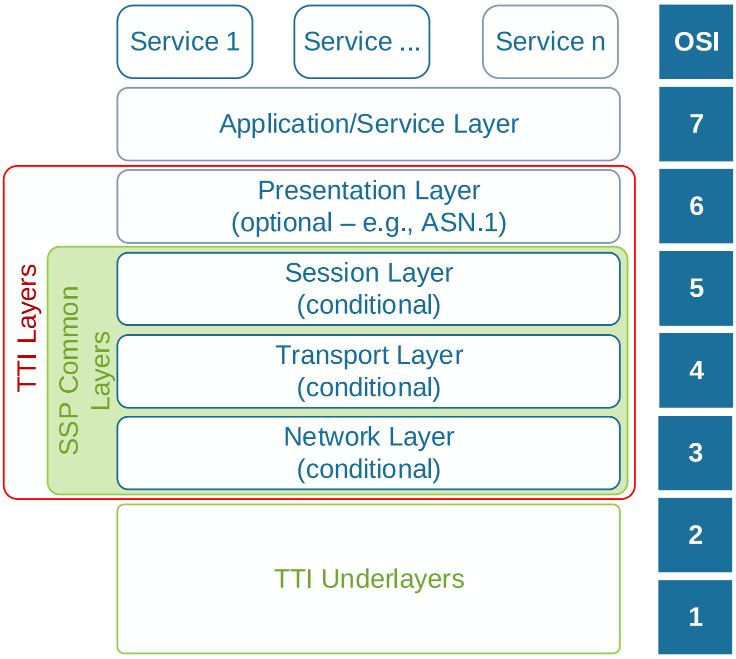

Figure 5.3: TTI layers represented in the OSI layer model

5.2.2 TTI underlayers

The TTI interacts with the services to test and shall be able to convey the SCL packets supporting the ETSI TS 103 999-1 [3], ETSI TS 103 999-2 [4], ETSI TS 103 666-1 to ETSI TS 103 666-4 [1], [2], [i.1] and [i.2]. The TTI underlayers shall support the same requirements as the ones driving the SCL layers and defined in ETSI TS 103 666-1 [1], clause 8.3.

The TTI underlayer shall be independent of any physical layer. With using TCP/IP, the support of all requirements expressed in ETSI TS 103 666-1 [1], clause 8.3 can be guaranteed by ensuring a stable and versatile connection between the SUT and the TT.

Any binding layer connecting any physical layer and the IP layer is convenient. The generic standards of IP binding for conveying IP packets on given physical layer is listed in the Annex G of the present document as example.

NOTE: No standard IP binding technology is required even a proprietary one can be used as soon as it is capable to convey IP packets from/to the terminal.

5.2.3 TTI underlayer server

The TTI underlayer (TTI_UL) related to the interface of a TT uses a stream based on TCP/IP technology via IP binding, supporting the transport of IP packets.

The TTI_UL may require the support of the TLS protocol as defined in IETF RFC 8446 [9].

The terminal hosting TTI shall establish a stream socket independently of the IP version (version 4 or 6) on a given port which is defined from the TTI_UL server URL.

5.2.4 TTI UL client/server connection

The TTI server represents the principle shown in figure 5.3. Figure 5.4 illustrates the client/server connection between the TT and the TTI_UL server in the terminal.

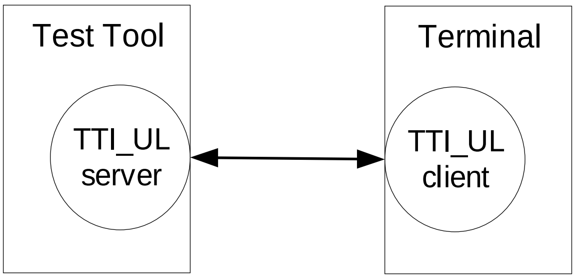

Figure 5.4: TTI_UL server in the Test Tool

The URL of the TTI_UL server is either provided by a MQTT broker defined by the TTM or the OEM. The TTI server URL may propose the support of a secure TCP session based on TLS as defined in IETF RFC 8446 [9].

Figure 5.5 illustrates the principle of TTI_UL server/client discovering.

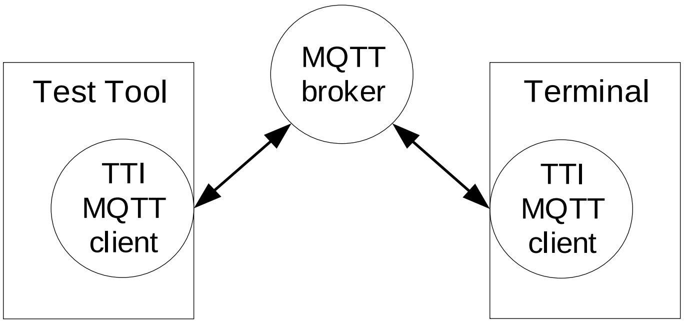

Figure 5.5: TTI_UL server/client discovering

Both TT and terminal shall connect a MQTT client to an agreed MQTT broker as defined in [8]. The MQTT connection requires the use of TLS with a mutual authentication of the broker and the client. This mutual authentication requires the exchange of digital certificates (and their associated private keys) provisioned into the terminal and the TT.

The steps leading to connect the TTI_UL client to the TTI_UL server are the following:

1) Both TTI MQTT clients expose their client identifier after a successful connection to the MQTT broker.

2) The TTI MQTT client of the terminal subscribes on a topic for getting the URL of the TTI_UL server.

3) The TTI MQTT client of the TT publishes the TTI_UL server URL using a syntax as defined in IETF RFC 3986 [11] on the above topic.

4) The TTI MQTT client of the terminal gets the URL of the TTI_UL server and the TTI_UL client connects it.

Any terminal and TT sharing the same MQTT broker may be connected. The conditional access to a given terminal from a TT leads to share a secret between both entities only. The sharing of this secret depends on the capabilities of out-of-band communication of the terminal (e.g. user interface, a button, etc.).

5.2.5 TTI MQTT certification paths

5.2.5.1 Overview

A TTI certification path for MQTT client authentication shall be prepared to consider the following:

- The CI issuers of both MQTT broker and MQTT clients which are respectively CI_{MB} and CI_{MC} may be different.
- No credentials from a MQTT client (e.g. CERT CI_{ULC}.ECDSA certificate) shall be provisioned into another MQTT client.

The protection of private keys associated to all public keys within these certificates are under the responsibility of each party. The means for protecting the keys are proprietary.

Figure 5.6 illustrates the x509 certification paths involved in the TTI interface.

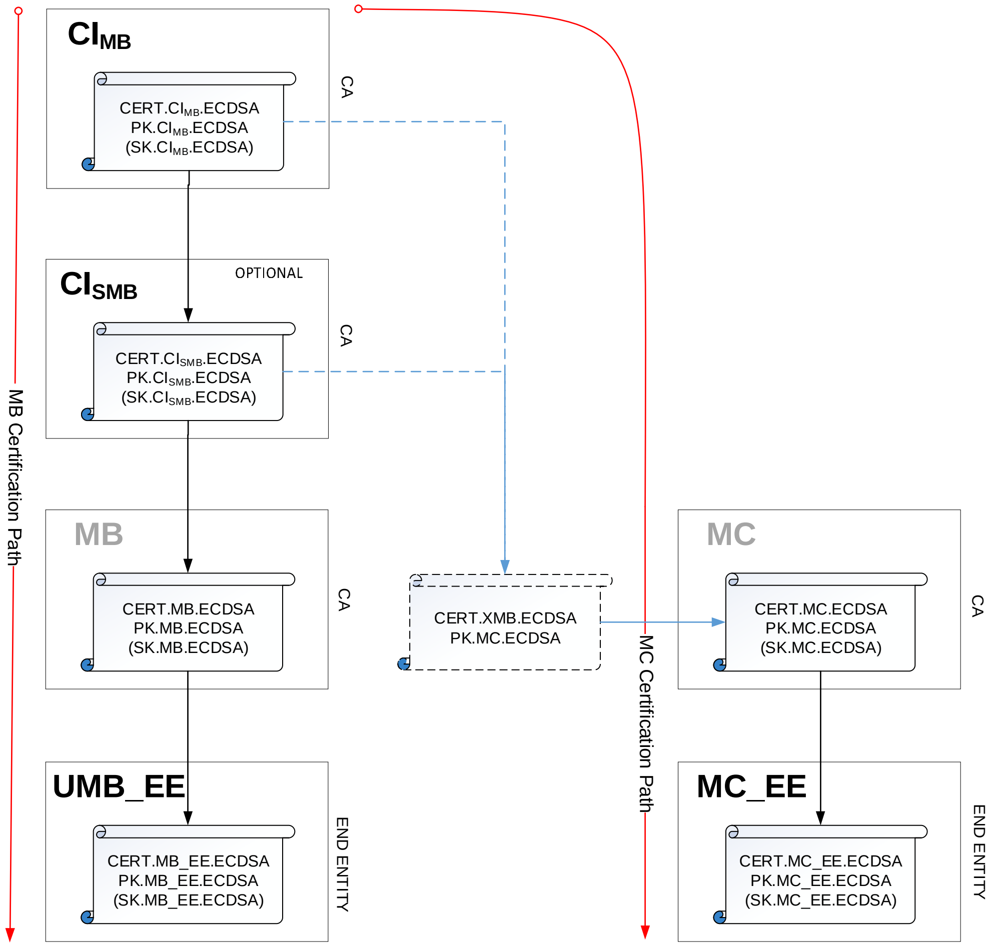

Figure 5.6: MQTT TTI certification paths

A cross certificate CERT.XMB.ECDSA signed by CI_{MB} is provisioned into all MQTT clients (TT and terminal). This cross certificate allows the verification of a certification path from CERT.CI_{MB}.ECDSA already provisioned into the TTI MQTT clients. An intermediate CA as CERT.CI_{SMB}.ECDSA may be inserted within the certification path.

A certificate CERT.TT_OEM.ECDSA, endorsing the TTM or the OEM, is generated on request as shown on figure 5.8.

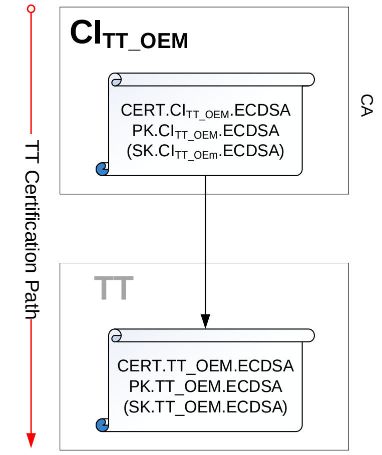

Figure 5.7: TTM/OEM certification path

Figure 5.8 illustrates the flow for requesting a cross-certificate from the TTI MQTT broker.

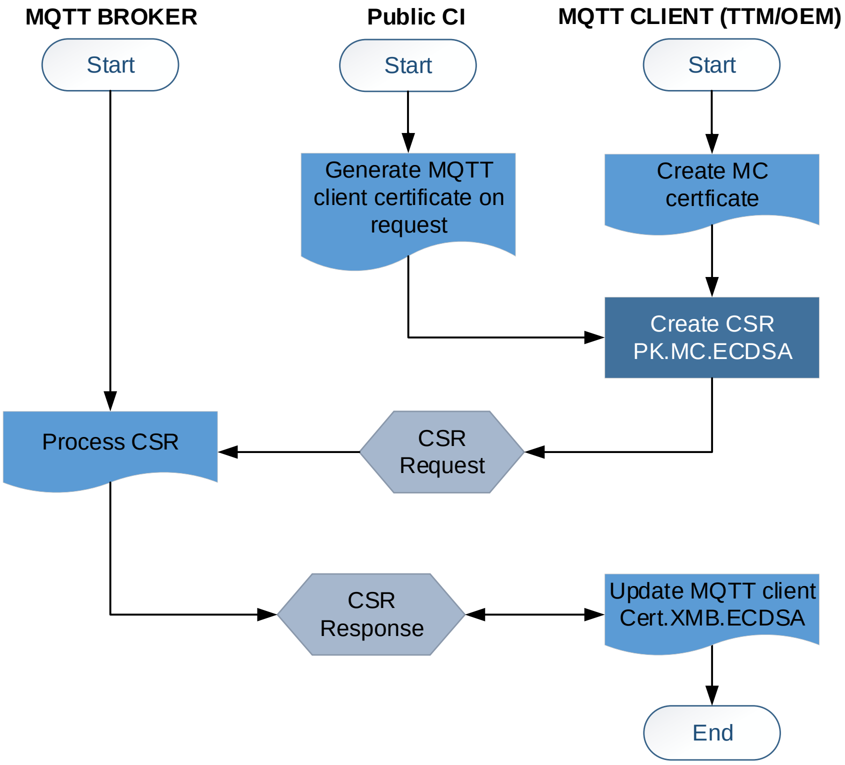

Figure 5.8: Cross certificate request (example)

The procedure is performed as follow and could be automated:

- The TTM or OEM requests a certificate CERT.TT_OEM.ECDSA from a public CI agreed by the OEM.
- The TTM/OEM generates/requests a certificate CERT.MC.ECDSA from its CI (CERT.CI$_{MC}$.ECDSA).
- The TTM/OEM creates a Certificate signing request (CSR), signed by SK.TT_OEM.ECDSA (private key associated to the PK.TT_OEM.ECDSA public key), for getting a cross certificate from the MQTT broker CI (CERT.CI$_{MB}$.ECDSA). The CSR contains the certification path to CERT.TT_OEM.ECDSA to authenticate the MQTT client. The CSR contains the information (i.e. PK.MC.ECDSA) for creating the CERT.XMB.ECDSA certificate allowing to verify the MC certification path as defined in figure 5.6.
- The TTM or OEM sends by any means (standard or not) this CSR to the MQTT broker CI.

- The MQTT broker CI generates a cross certificate CERT.XMB.ECDSA containing as subject public key PK.MC.ECDSA.
- The MQTT broker CI sends back the CSR response embedding the cross-certificate CERT.XMB.ECDSA to the TTM/OEM.
- The TTM/OEM provisions their TT or terminal with the CERT.XMB.ECDSA cross-certificate.

5.2.5.2 TTI MQTT certificate policy

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:

```
-- ASN1START

id-tti OBJECT IDENTIFIER ::= {itu-t (0) identified-organization (4) etsi (0) smart-secure-platform
(3834) part2 (1) }

id-mb-role OBJECT IDENTIFIER ::= {id-tti role (0)}

id-mb-role-ci OBJECT IDENTIFIER ::= { id-mb-role ci (0)}
id-mb-role-subordinate-ci OBJECT IDENTIFIER ::= { id-mb-role-ci subordinate-ca (0)}

id-mb-role-mb OBJECT IDENTIFIER ::= {id-mb-role-subordinate-ci mb (0)}
id-mb-role-xmb OBJECT IDENTIFIER ::= {id-mb-role-subordinate-ci xmb (1)}
id-mb-role-mc OBJECT IDENTIFIER ::= { id-mb-role-xmb mc (0)}

id-mb-role-mb-ee OBJECT IDENTIFIER ::= { id-mb-role-mb ee(0)}
id-mb-role-mc-ee OBJECT IDENTIFIER ::= { id-mb-role-mc ee(0)}

-- ASN1STOP
```

5.2.5.3 Certification path verification

All MQTT clients shall verify the certification path received by each other according to the IETF RFC 5280 [12].

5.2.5.4 TTI MQTT parameters

5.2.5.4.1 TTI MQTT broker connection parameters

The following parameters shall apply for connecting a TTI MQTT client to a MQTT broker:

- User Name Flag: 0
- Password Flag: 0 (the client authentication is performed with digital certificates)
- Will Retain: 0
- Will QoS: 1
- Will Flag: 0
- Clean Start: 0
- QoS: 2
- Retain flag: false

5.2.5.4.2 TTI MQTT client identifiers

The MQTT client identifier is computed from the canonical representation of a UUID version 5 as defined in IETF RFC 4122 [7] in only keeping the authorized symbol as defined in OASIS Standard [8]. The following example illustrates a valid canonical UUID for a client identifier: 4ebd0208-8328-5d69-8c44-ec50939c0967.

The URN definition leading to the UUID computation is defined as follow:

MQTT client identifier URN := urn:oem_domain_name:part_number:serial_number

Where:

1) urn:oem_domain_name is the NID (e.g. oemTool.com).

2) part_number:serial_number: is the NSS in which:

    - The part_number represents the Part Number defined by the OEM of the terminal or the TT.

    - The serial_number is the serial number of the equipment (terminal or TT) in the scope of the part number.

5.2.5.4.3 TTI MQTT topic name

The MQTT client topic for getting the TTI UL URL is defined as follow: /geturl/client_identifier.

The client_identifier is defined in the clause 5.2.5.4.2 of the present document.

As example, the following topic name is valid: /geturl/4ebd0208-8328-5d69-8c44-ec50939c0967

5.2.5.4.4 TTI MQTT payload

The payload conveyed between two TTI MQTT client for the defined topic name shall support the JSON syntax as defined in IETF RFC 7159 [10]. The payload for conveying the TTI_UL server URL is defined as follow:

{url:"tti:xxx:port"} or {url:"ttis:xxx:port"}

Where:

- tti defines a simple TCP connection between the TTI_UL server and client
- ttis defines a TLS over TCP connection between the TTI_UL server and client
- xxx represents the IP address or its DNS symbolic name
- port: the port number on which the TTI_UL server listen

5.2.6 Certificate description

5.2.6.1 Certificates common fields

Table 5.1 describes the basic certificate fields for all certificates in the present document follows X.509 v3 certificate format as defined in IETF RFC 5280 [12].

Table 5.1: Certificates common fields

<table>
<tr><th>Field</th><th colspan="2">Value Description</th></tr>
<tr><td rowspan="9">tbsCertificate</td><td colspan="2">Data to be signed</td></tr>
<tr><th>Parameter Field</th><th>Value Description</th></tr>
<tr><td>Version</td><td>Integer value of 2 denoting version 3.</td></tr>
<tr><td>serialNumber</td><td>Positive integer value assigned by the issuer to identify this certificate.</td></tr>
<tr><td>Signature</td><td>Identifier of signature algorithm used by the issuer to sign this certificate.
This value shall be the same as the one of 'signatureAlgorithm' field.</td></tr>
<tr><td>Issuer</td><td>Distinguished Name (DN) of the entity that has signed and issued this certificate. The value is defined as X.501 [21] type name.</td></tr>
<tr><td>Validity</td><td>Certificate validity period.</td></tr>
<tr><td>Subject</td><td>Distinguished Name (DN) of the entity associated with the public key in this certificate.</td></tr>
<tr><td>subjectPublicKeyInfo</td><td>Value of the public key and algorithm with which the key is used subjectPublicKeyInfo.algorithm shall be 'AlgorithmIdentifier' defined in clause 5.2.5.3. subjectPublicKeyInfo.subjectPublicKey shall be the value of public key coded as defined in IETF RFC 5480 [14].</td></tr>
<tr><td>signatureAlgorithm</td><td colspan="2">Identifier of signature algorithm used by the issuer to sign this certificate.
This value shall be the same as the one of 'signature' field.</td></tr>
<tr><td>signatureValue</td><td colspan="2">Digital signature computed over ASN.1. DER encoded tbsCertificate using digital signature algorithm.</td></tr>
</table>

5.2.6.2 Extension fields for Certificates

The following extension fields defined in IETF RFC 5280 [12] are considered:

- **Authority key identifier** (IETF RFC 5280 [12], section 4.2.1.1): All the certificate except for CI certificate shall contain the extension for authority key identifier.

- **Subject key identifier** (IETF RFC 5280 [12], section 4.2.1.2): 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.

- **Key usage** (IETF RFC 5280 [12], section 4.2.1.3): For a certificate used for verifying its subject certificate, keyCertSign (bit 5) shall be asserted to the key usage extension field of the certificate.

- **Certificate polices** (IETF RFC 5280 [12], section 4.2.1.4): Each certificate shall have the appropriate value of the extension for certificate policies.

- **SubjectAltName** (IETF RFC 5280 [12], section 4.2.1.6): A certificate may have the extension for subjectAltName.

- **Basic constraints** (IETF RFC 5280 [12], section 4.2.1.9): For any CA or subordinate CA certificate, the value of the extension for basic constraints shall be asserted.

5.2.6.3 Algorithm identifiers and parameters

This clause provides the value to be set in 'AlgorithmIdentifier' structure contained in 'subjectPublicKeyInfo', 'signature', and 'signatureAlgorithm' fields of the certificates for each of cryptographic algorithms used in the present document.

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 [14];

    - for SM2 digital signature algorithm, "iso(1) standard(0) digital-signature-with-appendix(14888) part3(3) algorithm(0) sm2(21)" as defined in ISO/IEC 14888-3 [16].

  - 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 [14].

- '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 [17];
  - 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 [17];
  - for NIST P-256: "iso(1) member-body(2) us(84
  - (0) ansi-X-9-62(10045) curves(3) prime(1) secp256v1(7)" as defined in IETF RFC 5480 [14];
  - for NIST P-384: "iso(1) identified-organization(3) certicom(132) curve(0) secp384r1(34)" as defined in IETF RFC 5480 [14].

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) ecdsa-with-SHA2(3) ecdsa-with-SHA256(2)" as defined in IETF RFC 5758 [15];
  - for ECDSA-with-SHA384: "iso(1) member-body(2) us(840) ansi-X9-62(10045) signatures(4) ecdsa-with-SHA2(3) ecdsa-with-SHA384(3)" as defined in IETF RFC 5758 [15];
  - 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 [16].
- '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 [13], section 3.2.

5.3 Test Tool SCL network

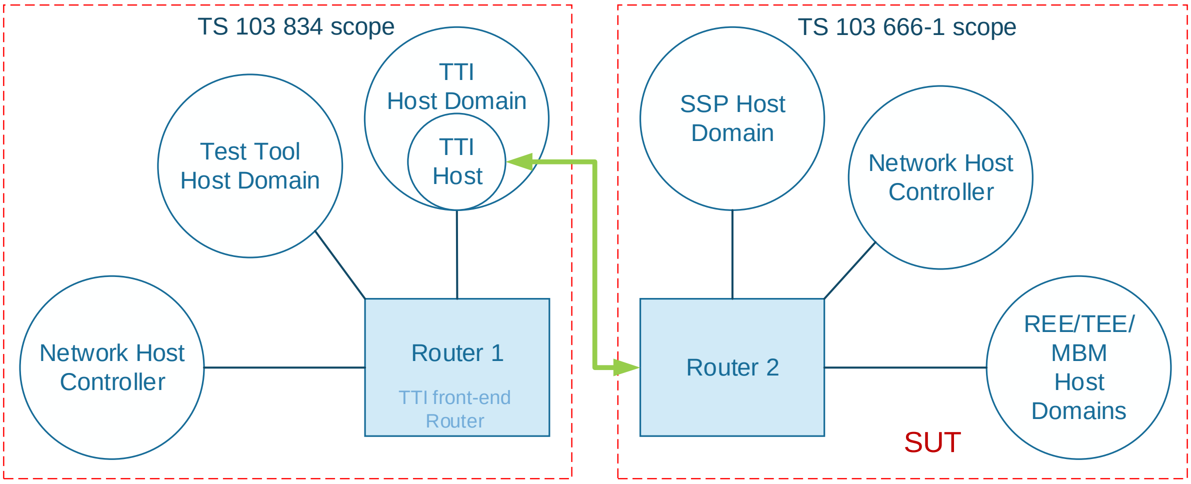

Figure 5.9: Router abstraction

As figure 5.9 exposes, the data handling on the TT is done in two independent SCL networks:

- A SCL network dedicated to the tests and hosting two host domains (TTI Host Domain and TT Host Domain) and its Network controller host.
- The SCL network under test as defined in the ETSI TS 103 666-1 [1].

The two SCL networks are isolated and a conditional tunnelling of SCL packets is allowed to the Router 2.

In order to fulfil the requirements from ETSI TS 103 666-1 [1], despite the restriction from ETSI TS 103 666-1 [1], clause 8.2, not to allow services not hosted within the SSP host, the TT Host Domain is impersonated as an SSP host domain in order to test the services within the other hosts not in the SSP host domain.

Router 1 acts as a communication front-end of the TT Host. Router 2 is isolated from an insecure environment. The TTI Host allows the bridging to Router 2 from any TT Host associated to an authenticated TT Accessor.

5.4 TTI platform

5.4.1 TTI valid platform

Figure 5.10 illustrates a TTI valid platform.

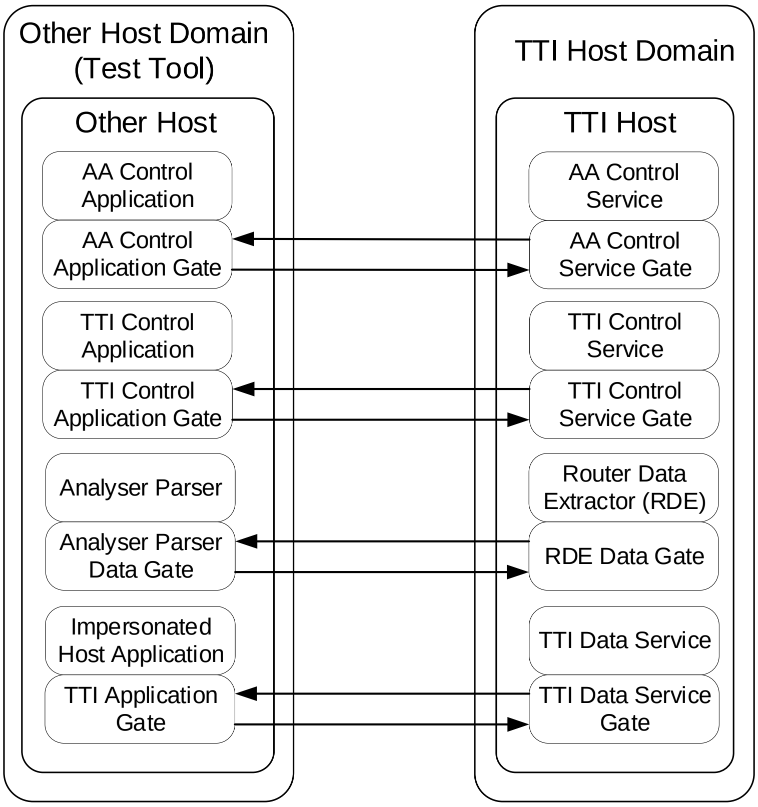

Figure 5.10: TTI valid platform

In addition to the mandatory core gates (administration, loopback, link and identity), the TTI Host shall embed at least four gates:

- the Accessor Authentication Service gate;
- the TTI Control Service gate;
- the TTI Data Service gate;
- the RDE Data Service gate.

Via the AAS (Accessor Authentication Service), the Accessor Authentication Application can bootstrap the TTI Control Service to set the parameters for the router. The TTI Control Service allows to bootstrap the access to:

- The Router Data Extractor (RDE) Data Service gate, which copies the timestamped SCL packets from the router and encapsulates them into SCL packets (tunnelling). These packets are conveyed to the TTI Analyzer Data gate before they are received by the parser of the analyser.

- The TTI Data Service gate, for tunnelling SCL packets from/to an impersonated host.

The use of the AAS allows to control the access to authorized accessor for tests purposes only. Moreover, the AAS allows to setup secure SCL to a remote TT what enables the TT to act as a service operating on a cloud based infrastructure.

5.4.2 Router directives and data

Figure 5.11 illustrates the interaction between TTI services and Router 2, The figure exemplifies the handling of transfer directives and timestamped copies of SCL packets.

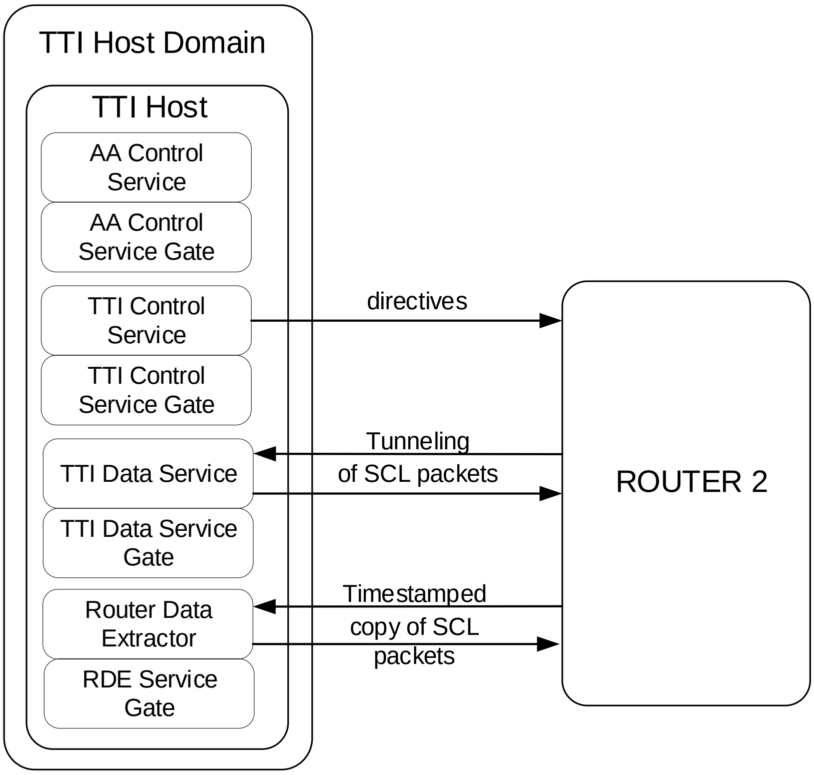

Figure 5.11: Router directives and data

5.4.3 TTI credential storage

Credentials (e.g. private keys) allowing the authentication of the AAS within the TTI Host shall be protected.

Two constraints may impact the implementation of the TTI Host:

- The AAS within the TTI Host may only control the access to services within that host.

- The TTI Host cannot be in the SSP host domain.

In consequence, the TTI Host Domain may either be part of the REE host domains or the TEE host domains (if available).

Nevertheless, during the authentication of the TT Accessor the cryptographic operations related to the accessor authentication may be delegated to a service within an SSP host.

Figure 5.12 illustrates the concept of delegation, in which the execution of the cryptographic operations is performed by a service of an SSP host as defined in ETSI TS 103 666-1 [1], clause 9.3.2. The computed shared secret is returned to the delegation application by using secure SCL pipes. The level of security to protect the credentials stored within the AAS in the TTI Host is equivalent to the level of security of the TTI Host Domain (e.g. TEE host domains). The computed shared secrets are ephemeral, and the security impacts related to their disclosure are less sensitive than a static private key associated to the accessor authentication. The level of security of a group of communicating entities is as high as the level of security of the weakest entity.

The entry point for the SCL network communicating with the hosts as defined in ETSI TS 103 666-1 [1], is the TTI Data Service as defined in the clause 6.3 of the present document. In consequence, the weakest security link is the TTI Data Service which is running on the terminal.

The delegation of the credential storage within a service of an SSP host as proposed by the figure 5.12 does not upgrade the global security of the TTI interface.

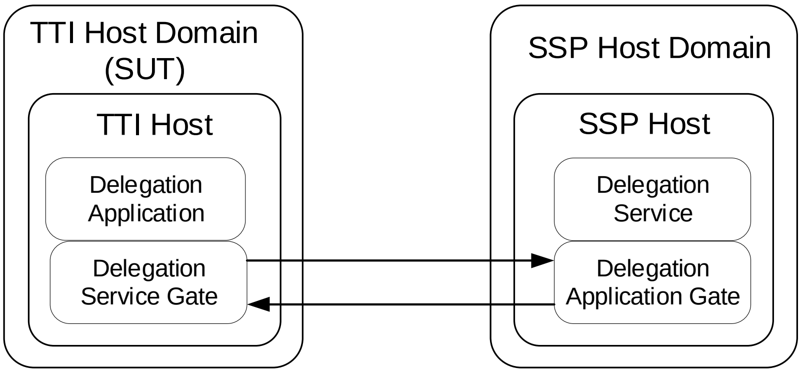

Figure 5.12: Delegation

5.5 TTI security perspectives

5.5.1 Identification of security perspectives

Dumping the SCL packets from a host and/or impersonating a host to emulate a host in a TT may lead to security breaches and to disclose sensitive data (see ResearchGate [i.3]). The access to the TT interface shall be enforced with established and secure methods. Thus, the TTI:

1) Reuses the principles of the ETSI TS 103 666-1 [1], standard in order to:

   - Shorten the learning phase for solution implementers.
   - Be able to test the conditional access to the TTI by reusing the tests descriptions defined in ETSI TS 103 999-1 [3].

2) Reuses SCL network as the infrastructure of communication between the TT and the TTI in the SUT.

3) Reuses a standard transport layer for carrying the SCL packets of the above SCL network infrastructure:

   - If the SCL is independent from a physical layer, then the TTI may be accessible via any physical layer available on the SUT capable to provide the minimum bandwidth required.

4) Does not mandate the support of a wired connection to a physical layer:

   - Some SUT (e.g. smart watches) may not provide any wired connection to interface with their physical layer. In such cases a TTI SCL underlayer mandating the support wired physical layer is not suitable.

5.5.2 TTI security requirements

The TTI shall fulfil the following security requirements:

- The access to the TTI shall be possible for authenticated accessors only.
- The TTI may request confidential data exchange to the TT.
- An access control shall allow to grant dumped SCL packets and services per host and per services.

5.6 TTI certificates

5.6.1 TTI certification paths

A TTI certification path shall be prepared to consider the following:

- The CI issuers of both OEM and TTM which are respectively CI_{AAS} and CI_{AAA} may be different.
- No credentials from the test maker (e.g. CERT CI_{AAA}.ECDSA certificate) shall be provisioned into the TTI Host for supporting the accessor authentication with the TTI Host AAS.

The protection of private keys associated to all public keys within these certificates are under the responsibility of each party. The means for protecting the keys are proprietary.

Figure 5.13 illustrates the x509 certification paths involved in the TTI interface.

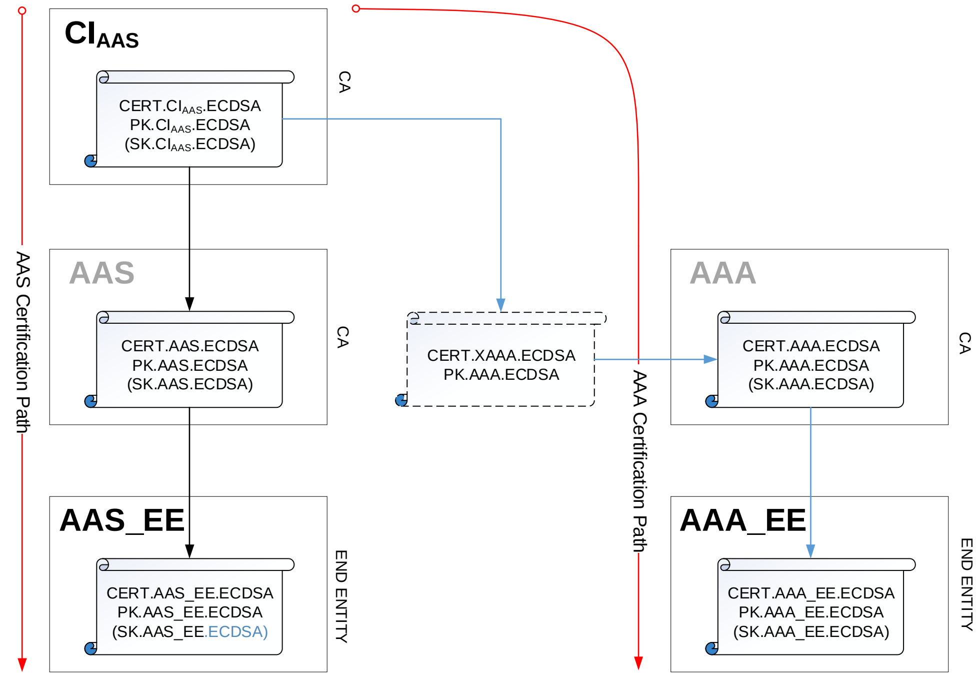

Figure 5.13: TTI certification paths

A cross certificate CERT.XAAA.ECDSA signed by CI_{AAS} is provisioned by the test maker into the TT. This cross certificate allows the verification of a certification path from CERT.AAS.ECDSA already provisioned into the AAS of the TTI host.

A certificate CERT.TT.ECDSA endorsing the TTM is generated on request as shown on figure 5.14.

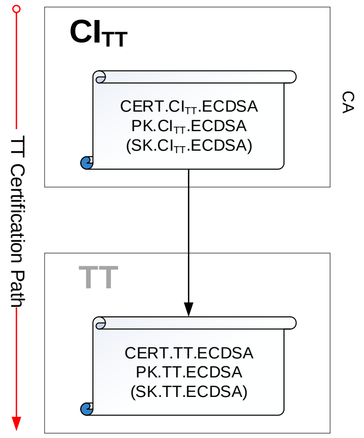

Figure 5.14: TTM certification

Figure 5.15 illustrates the flow for requesting a cross certificate from the OEM.

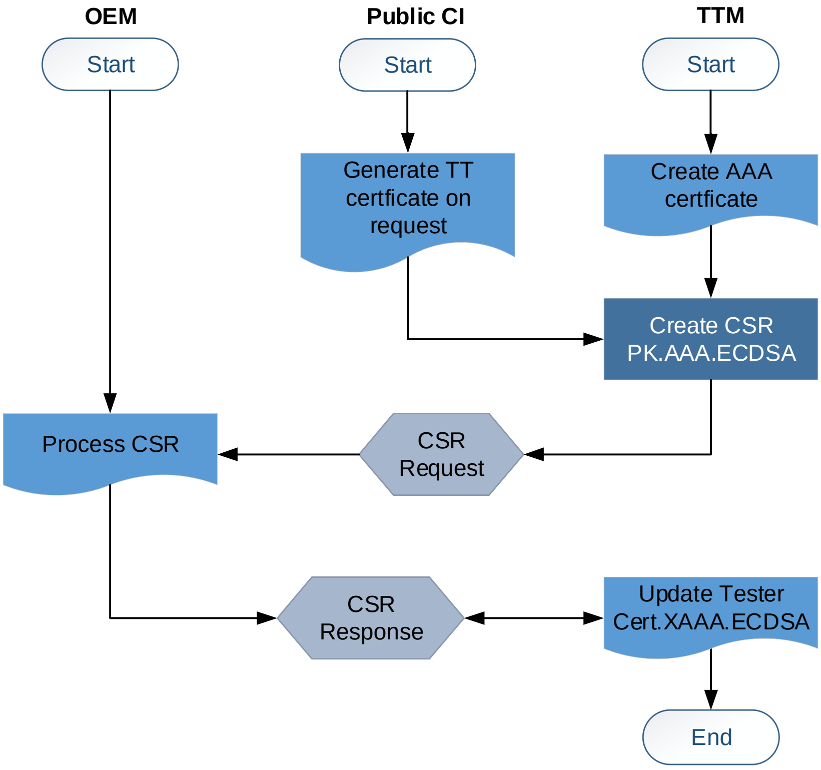

Figure 5.15: Cross certificate request (example)

The procedure is performed as follow and could be automated:

- The TTM requests a certificate CERT.TT.ECDSA from a public CI agreed by the OEM.
- The TTM generates/requests a certificate CERT.AAA.ECDSA from its CI (CERT.CI$_{AAA}$.ECDSA).
- The TTM creates a Certificate signing request (CSR) for getting a cross certificate from the OEM CI (CERT.CI$_{AAS}$.ECDSA). The CSR contains the certification path to CERT.TT.ECDSA to authenticate the TTM.
- The TTM sends by any means (standard or not) this CSR to the OEM.
- The OEM generates a cross certificate CERT.XAAA.ECDSA containing as subject public key PK.AAA.ECDSA.

- The OEM sends back the CSR response embedding the cross-certificate CERT.XAAA.ECDSA to the TTM.
- The TTM provisions the TT with the cross certificate.

5.6.2 Certification path verification

The certification path verification shall support the requirement as defined in clause 5.2.5.3 of the present document.

5.6.3 Certificate policies

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:

```
-- ASN1START

id-mb-role OBJECT IDENTIFIER ::= {id-tti role (0)}

id-mb-role-ci OBJECT IDENTIFIER ::= { id-mb-role ci (0)}
id-mb-role-subordinate-ci OBJECT IDENTIFIER ::= { id-mb-role-ci subordinate-ca (0)}

id-mb-role-mb OBJECT IDENTIFIER ::= {id-mb-role-subordinate-ci mb (0)}
id-mb-role-xmb OBJECT IDENTIFIER ::= {id-mb-role-subordinate-ci xmb (1)}
id-mb-role-mc OBJECT IDENTIFIER ::= { id-mb-role-xmb mc (0)}

id-mb-role-mb-ee OBJECT IDENTIFIER ::= { id-mb-role-mb ee(0)}
id-mb-role-mc-ee OBJECT IDENTIFIER ::= { id-mb-role-mc ee(0)}

id-aas-role OBJECT IDENTIFIER ::= {id-tti role (1)}

id-aas-role-ci OBJECT IDENTIFIER ::= { id-aas-role ci (0)}

id-aas-role-aas OBJECT IDENTIFIER ::= {id-aas-role-ci aas (0)}
id-aas-role-xaas OBJECT IDENTIFIER ::= {id-aas-role-ci aas (1)}
id-aas-role-aaa OBJECT IDENTIFIER ::= { id-aas-role-xaas aaa (0)}

id-aas-role-aas-ee OBJECT IDENTIFIER ::= { id-aas-role-aas ee(0)}
id-aas-role-aaa-ee OBJECT IDENTIFIER ::= { id-aas-role-aaa ee(0)}

-- ASN1STOP
```

6 TTI services

6.1 TTI Accessor Authentication Service

6.1.1 Requirements

The TTI Accessor Authentication Service is the service in the TTI Host responsible of authenticating external entities accessing resources on the TTI Host.

The TTI Accessor Authentication Service shall operate similar to the Accessor Authentication Service in the SSP, defined in ETSI TS 103 666-1 [1], clause 6.13. Where the TT becomes a member of the accessor repository, allowed to use of the service within a TTI Host outside the SSP host domain.

6.1.2 Test Tool (TT) Accessor Authentication

The TT Accessor Authentication supported by the Accessor Authentication Service within the TTI Host shall support the protocol as defined in ETSI TS 103 666-1 [1], clause 9.4.

6.1.3 Access to the TTI Host Services

The access of the TTI Host Services as defined in clause 5.4 of the present document are granted by using the AAS-OP-ACCESS-SERVICE-Service-Command as defined in ETSI TS 103 666-1 [1], clause 6.13.5.6.

6.2 TTI Control Service

6.2.1 Overview

This service aims to manage the constraints and the behaviour of the Router Data Extractor interfacing the Router 2.

6.2.2 Administrative operations

The TTI Control Service supports the following administrative operations:

- retrieving the capabilities of the TTI Control Service (i.e. TTI-OP-GET-CAPABILITIES-Service-Command);
- updating the directives of a TTI Data Service related to the Router Data Extractor in charge to record the SCL packet traffic.

6.2.3 TTI Control Service access

6.2.3.1 Access rights

The TTI Control Service access rights allow to define the availability of requested operations.

```
-- ASN1START

eTTIAccessRight-MBMHostPacketRecordAllowed AccessorRights ::=    { eRight-Bit1 }
eTTIAccessRight-MBMHostPacketInjectionAllowed AccessorRights ::= { eRight-Bit2 }
eTTIAccessRight-MBMHostImpersonationAllowed AccessorRights ::=   { eRight-Bit3 }
eTTIAccessRight-SSPHostImpersonationAllowed AccessorRights ::=   { eRight-Bit4 }
eTTIAccessRight-APDUGateAccessAllowed AccessorRights ::=         { eRight-Bit5 }
eTTIAccessRight-UpdateACLAllowed AccessorRights ::=              { eRight-Bit6 }

-- ASN1STOP
```

Where:

- **eTTIAccessRight-MBMHostPacketRecordAllowed**: this right indicates that, in addition to the permissions required to access the resource, the impersonated host is allowed to record the SCL packets conveyed from/to the MBM host. The RDE data service shall not copy the SCL packet from/to the MBM host.
- **eTTIAccessRight-MBMHostPacketInjectionAllowed**: this right indicates that, in addition to the permissions required to access the resource, the impersonated host is allowed to send/receive the SCL packets from/to the MBM host. The MBM host shall not include the impersonated host is in whitelist.
- **eTTIAccessRight-MBMHostImpersonationAllowed**: this right indicates that, in addition to the permissions required to access the resource, the impersonated host is allowed to impersonate the MBM host.
- **eTTIAccessRight-SSPHostImpersonationAllowed**: this right indicates that, in addition to the permissions required to access the resource, the impersonate host is allowed to impersonate the SSP host.
- **eTTIAccessRight-APDUGateAccessAllowed**: this right indicates that, in addition to the permissions required to access the resource, the impersonated host is allowed to send, receive and record SCL packet from/to a APDU service/application gate of a host. The host shall not accept the opening of a pipe session to the APDU service gate.
- **eTTIAccessRight-UpdateACLAllowed**: this right allows the update of the access control list of the TTI control service.

Table 6.1 illustrates the applicability of the rights to dedicated TTI commands.

Table 6.1: Applicability of rights

Command	Right					
	eTTIAccessRight-MBMHostPacketRecordAllowed	eTTIAccessRight-MBMHostPacketInjectionAllowed	eTTIAccessRight-MBMHostImpersonationAllowed	eTTIAccessRight-SSPHostImpersonationAllowed	eTTIAccessRight-APDUGateAccessAllowed	eTTIAccessRight-UpdateACLAllowed
TTI-OP-GET-CAPABILITIES-Service-Command						
TTI-ADMIN-IMPERSONATE-Service-Command			●	●		
TTI-ADMIN-UPDATE-ACL-Service-Command						●

In order to run the test cases defined in ETSI TS 103 999-1 [3] and ETSI TS 103 999-2 [4] the relevant access rights shall be granted to the TT accessor.

6.2.3.2 Directives

6.2.3.2.1 Directives transfer

The TTI Control Service shall forward the directives related to rights, as defined in clause 6.2.3.1, to the Router 2 and the SUT host network controller. This rights transfer is implementation dependent.

The SUT host network controller shall broadcast the following event concerned by the directives. Table 6.2 defines the events.

Table 6.2: Directive Events

Value	Event
'20'	EVT_TTI_DIRECTIVES

6.2.3.2.2 EVT_TTI_DIRECTIVES

This event shall be sent by the link service gate of the SUT host network controller to all connected hosts concerned by the TTI directives as defined in clause 6.2.3.1. All recipients of the EVT_TTI_DIRECTIVES shall enforce the directives.

This event has the following parameter.

```
-- ASN1START

TTI_Directives ::= SEQUENCE
{
aImpersonnatedHost UUID, -- Host identifier of the impersonated host
aDirectives AccessorRights  -- Directives as defined in clause 6.2.3.1
}
-- ASN1STOP
```

6.2.3.3 Primitives

6.2.3.3.1 TTI-OP-GET-CAPABILITIES-Service-Command

With the command TTI-OP-GET-CAPABILITIES-Service-Command, a TTI Control Service Application requests the TTI Control Service to get granted the capabilities of the TTI Control Service.

```
-- ASN1START

TTI-OP-GET-CAPABILITIES-Service-Command ::= [PRIVATE 16] SEQUENCE
{
}
-- ASN1STOP
```

This command has no parameters.

If the capabilities can be granted, the TTI Control Service shall send a response including an eTTI-OK.

```
-- ASN1START

TTI-OP-GET-CAPABILITIES-Service-Response-Parameter ::= [PRIVATE 16] SEQUENCE
{
    aVersion Version  -- Release of the TTI service
}

TTI-OP-GET-CAPABILITIES-Service-Response ::= [PRIVATE 16] SEQUENCE
{
    aTTI-Service-Response TTI-Service-Response DEFAULT eTTI-OK,
    aParameter TTI-OP-GET-CAPABILITIES-Service-Response-Parameter OPTIONAL
}
-- ASN1STOP
```

Where:

- **aVersion:** major and minor release version supported by the file system Control Service gate;

6.2.3.3.2 TTI-ADMIN-IMPERSONATE-Service-Command

With the command TTI-ADMIN-IMPERSONNATE-Service-Command, a TTI Control Service Application may impersonate a host.

```
-- ASN1START

TTI-ADMIN-IMPERSONATE-Service-Command ::= [PRIVATE 17] SEQUENCE
{
    aFirmwareFamilyID UUID,  -- Identifier of firmware family of the host to impersonate
    aHostDomainID UUID -- Host domain identifier of the host to impersonate
}
-- ASN1STOP
```

This command has the following parameters:

- **aFirmwareFamilyID:** contains the identifier of the firmware family of the host to impersonate;
- **aHostDomainID:** contains the identifier of the host domain in which the impersonated host will belong to.

If the TTI Control Service Application can impersonate a host, the TTI Control Service shall send a response including an eTTI-OK.

```
-- ASN1START

TTI-ADMIN-IMPERSONATE-Service-Response ::= [PRIVATE 17] SEQUENCE
{
    aTTI-Service-Response TTI-Service-Response DEFAULT eTTI-OK
}
-- ASN1STOP
```

The TTI-Service-Response shall contain at least one the following parameters:

```
-- ASN1START

TTI-Service-Response ::= ENUMERATED
{
    eTTI-OK (0),  -- no error
    eTTI-E-CMD-PAR-UNKNOWN (2),  -- unknown or illegal command parameter
    eTTI-E-NOK (3)  -- the command has failed
}
-- ASN1STOP
```

6.2.3.3.3 TTI-ADMIN-UPDATE-ACL-Service-Command

With the command TTI-ADMIN-UPDATE-ACL-Service-Command, an TTI control application requests the TTI control service to update the access control.

The accessor updating the access control shall have the eTTIAccessRight-UpdateACL right on that resource.

```
-- ASN1START

TTI-ADMIN-UPDATE-ACL-Service-Command ::= [PRIVATE 18] SEQUENCE
{
    aACL SET OF AccessControl -- New access control
}

-- ASN1STOP
```

This command has the following parameters:

- aACL: the new access control list of the node.

When the request is successful, then TTI control service shall include eTTI-OK in the response.

```
-- ASN1START

TTI-ADMIN-UPDATE-ACL-Service-Response ::= [PRIVATE 18] SEQUENCE
{
    aTTI-Service-Response TTI-Service-Response DEFAULT eTTI-OK
}

-- ASN1STOP
```

6.2.4 Commands

The TTI Control Service gate supports the following commands.

```
-- ASN1START

TTI-SERVICE-GATE-Commands ::= [APPLICATION 2] CHOICE
{
    aTTI-OP-GET-CAPABILITIES-Service-Command TTI-OP-GET-CAPABILITIES-Service-Command,
    aTTI-ADMIN-IMPERSONATE-Service-Command TTI-ADMIN-IMPERSONATE-Service-Command,
    aTTI-ADMIN-UPDATE-ACL-Service-Command TTI-ADMIN-UPDATE-ACL-Service-Command
}

-- ASN1STOP
```

All the commands are described in clause 6.2.3.3.

6.2.3 Responses

The gate shall support the responses defined as follows.

```
-- ASN1START

TTI-SERVICE-GATE-Responses ::= [APPLICATION 1] CHOICE
{
    aTTI-OP-GET-CAPABILITIES-Service-Response TTI-OP-GET-CAPABILITIES-Service-Response,
    aTTI-ADMIN-IMPERSONATE-Service-Response TTI-ADMIN-IMPERSONATE-Service-Response,
    aTTI-ADMIN-UPDATE-ACL-Service-Response TTI-ADMIN-UPDATE-ACL-Service-Response
}

-- ASN1STOP
```

All the responses are described in clause 6.2.3.3.

6.3 TTI Data Service

6.3.1 Overview

This service is in charge to tunnel a SCL packets traffic from/to a TT Host which may impersonate a host as defined in clause 6.3.2.2 of the present document.

6.3.2 Principles

6.3.2.1 SCL packet tunnelling

Figure 6.1 is showing the encapsulation of SCL packets as used to communicate between TT Host and an SCL host B.

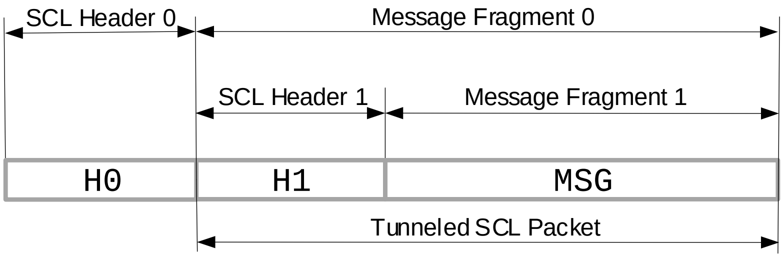

Figure 6.1: Encapsulation

Figure 6.2 illustrates the sending of an SCL packet, including the encapsulation into another SCL packet, its transfer via an opened SCL Tunnel to the TTI data service in charge to encapsulate/decapsulate before receiving/sending it to the Router 2.

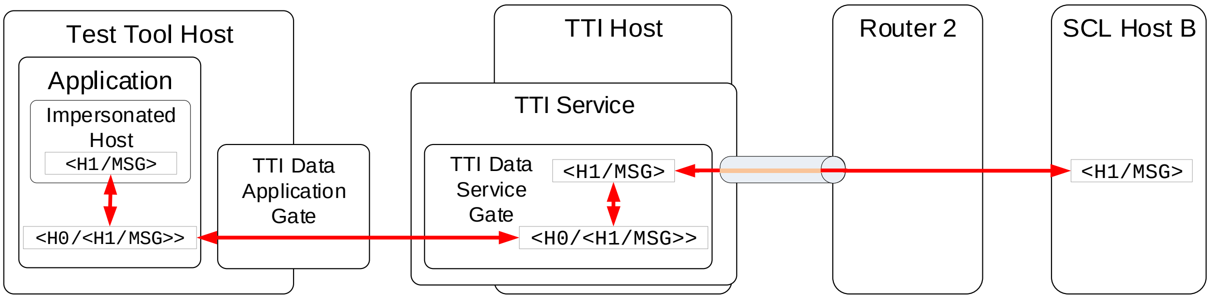

Figure 6.2: SCL packet tunnelling

The TTI Data Service gate is in charge to encapsulate/decapsulate SCL packets to be tunnelled between the SCL network of the SUT and the TT Host. An application of the TT Host emulates any host within the SCL network of the SUT system by using a SCL packet tunnelling.

If the impersonated host running in the Application of the TT Host sends an SCL packet to the SCL host B:

- The SCL packet (consisting of an SCL header and a message fragment) of the impersonated host, is encapsulated into another SCL packet by adding a new SCL header addressing the SCL Service gate and keeping the former SCL packer as message fragment.
- The encapsulated packet is sends via the TTI Data Application gate to the TTI Data Service gate within the TTI Host.
- The TTI Data Service gate of the TTI Host shall:
  - decapsulate the initial SCL packet from the SCL packet generated in the Application;
  - verify the SCL packet routing directives against the TT Accessor rights as defined in clause 6.1.2 of the present document;
  - forward the SCL packet to Router 2.

The means for the RDE data gate of the TTI Host for communicating with the Router 2 is implementation dependent.

If Router 2 sends an SCL packet:

- the TTI Data Service gate of the TTI Host shall:
  - encapsulate the SCL packet into another SCL packet by adding a new SCL header addressing the Application in the TT Host;
  - forward the encapsulated SCL packet via the TTI Application gate to the TT Host.

6.3.2.2 Host impersonation

The impersonation of a host is performed by the application emulating the impersonated host by using the following procedure:

- The TT Host shall initiate the impersonation of a host by using the TTI-ADMIN-IMPERSONNATE-Service-Command command as defined in clause 6.2.3.3.2.
- The successful acceptation of the impersonation directive sent by the TTI Control Service to the Router 2 in which the firmware family identifier given in aFirmwareFamilyID is associated to a host belonging the host domain identified via aHostDomainID. the Router 2 shall inform its network host controller that the impersonated host belongs to the host domain provided in aHostDomainID.
- The SCL packet tunnelling as defined in clause 6.3.2.1.
- The successful acceptation of the command LINK_REGISTER_HOST with the firmware identifier parameter given in aFirmwareFamilyID sent by the impersonated host.

The network host controller associated to the Router 2 shall register the impersonated host as belonging to the host domain provided in aHostDomainID.

This service shall filter the encapsulated SCL packets according to the rights of the TT Accessor as defined in the clause 6.1.2 of the present document.

6.3.3 TTI Data Service gate

6.3.3.1 Overview

This service does not support any commands or events.

The SCL packet encapsulating another SCL packet shall be configured as a data stream by setting the chaining bit (CB) of the HCP packet header to 0 (see ETSI TS 102 622 [18], clause 5.1).

The data acknowledgement mechanism (EVT_ADM_RECEIVED) and the credit-based data flow control (EVT_ADM_CREDIT) described in clause 8.5.3 in ETSI TS 103 666-1 [1] shall apply.

6.4 RDE Data Service

6.4.1 Overview

This service is in charge to:

- Filter the copied SCL packet flowing within the SCL network associate to the Router 2 (SUT) according to the rights associated to the TT Accessor as defined in clause 6.1.2 of the present document.
- Timestamp the taped SCL packets and encapsulate them into an RDE_EVT event.

6.4.2 RDE Data Service gate

6.4.2.1 About the RDE Data Service gate

This service does not support any commands. This gate supports the HCI presentation layer as defined in ETSI TS 102 622 [18], clause 5.2.

The data acknowledgement mechanism (EVT_ADM_RECEIVED) and the credit-based data flow control (EVT_ADM_CREDIT) described in ETSI TS 103 666-1 [1], clause 8.5.3 shall apply.

6.4.2.2 RDE Data Service gate Events

6.4.2.2.1 Overview

The gate supports the following event:

Table 6.3: RDE Service gate Events

Value	Event
'10'	EVT_RDE

6.4.2.2.2 RDE-Event

With the event RDE-Event, the RDE data service gate notifies the RDE data application gate that a SCL packet has been routed from Router 2. This event has the following parameter.

Table 6.4: RDE Parameter

Description	Length
Timestamp	8
copied SCL packet	N

Where:

- **Timestamp:** contains the relative time from the pipe session time for the RDE data service date of the issuance of the SCL packet from the Router 2. The value is an unsigned integer (64 bit) and the time unit is µs.
- **copied SCL packet:** contains the taped SCL packet.

6.5 Service identifiers

The provisions of GlobalPlatform VPP - Network Protocol [5], clause 3 shall apply.

Table 6.5 defines the URN for the additional gates defined in the present document, other than the ones referenced from GlobalPlatform VPP - Network Protocol [5]. All UUIDs are calculated using the version 5 of the UUID as specified in IETF RFC 4122 [7], using the domain name system namespace.

Table 6.5: Gates URN

Gate	NID	NSS	Pre-computed identifier
TTI control service gate	urn:etsi.org	TTI:ASN:TTI-control	09560b78-bed9-58b9-a5ff-6caa8384d556
TTI data service gate	urn:etsi.org	TTI:HCI.1:TTI-data	03040a72-7f68-58c8-bb57-d6f3e4c142d2
RDE data service gate	urn:etsi.org	TTI:HCI.1:RDE-data	fcb7bf93-a5de-5a3e-a1bb-ec6996052afa

The data acknowledgement mechanism (EVT_ADM_RECEIVED) and the credit-based data flow control (EVT_ADM_CREDIT) described in ETSI TS 103 666-1 [1] shall not apply unless otherwise specified in the gate description.

7 iSSP primary platform support

7.1 Scope

The iSSP Primary Platform (PP) and its Interface (PPI) are respectively defined in ETSI TS 103 666-2 [2], clauses 7 and 8.

The test descriptions related to the PPI are out of the scope of the present document. Nevertheless, the TTI shall support the tooling in charge to test the PPI.

The PPI exposes two interfaces:

- An API for accessing some services as defined in ETSI TS 103 666-2 [2], clause 7.3.
- An ABI for accessing the services as defined in ETSI TS 103 666-2 [2], clause 8.1.

The access to the PPI requires the use of a specific SSP host embedding a service interfacing the PPI and an application in the TT host.

7.2 Principle

Figure 7.1 illustrates the concept supporting the test of the PPI from the TT Host.

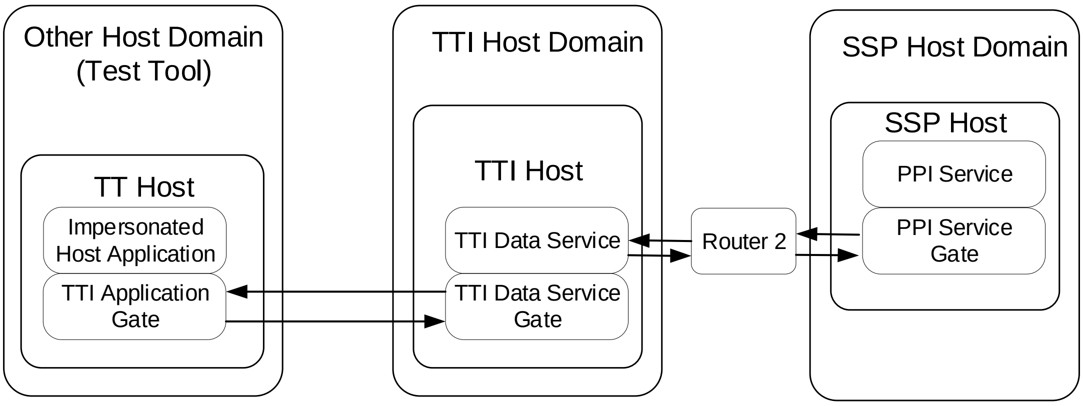

Figure 7.1: PPI service

An application within the TT Host of a TT Host Domain exposes an impersonated host to the SUT SCL network as defined in the clause 6.3.2 of the present document. Via the host impersonation, a host running TT appears in the SCL network of the SUT.

Figure 7.2 illustrates the equivalent platform allowing the test of the services accessible with the PPI.

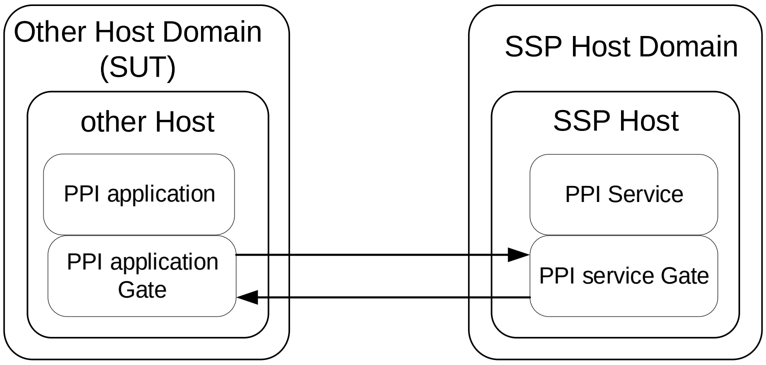

Figure 7.2: PPI service access

The PPI service may act as a means for calling remotely from the PPI application and PP service via its PPI. The definition of the PPI service is out of the scope of the present document.

7.3 TTI Data Service extension

The TT Data Service as defined in the clause 6.3 of the present document may be extended for supporting the additional events and directives related to [i.6], clause 5.6 in order to monitor:

- The signals sent/received by the MAIN process from the MGT service.
- The exceptions issued by the MAIN process.

This extension is out of the scope of the present document.

Annex A (informative): References on ETSI forge

A.1 ETSI forge repository for the TTI technical specification

- https://forge.etsi.org/rep/set/etsi-ts-103-834-part-1/tree/17.0.0.

A.2 License information

- https://forge.etsi.org/rep/set/etsi-ts-103-834-part-1/blob/17.0.0/LICENSE.

A.3 ASN.1 coding

The complete ASN.1 coding is available on ETSI forge.

- https://forge.etsi.org/rep/set/etsi-ts-103-834-part-1/tree/17.0.0/asn1.

A.4 UML code of figures

Table A.4-1 contains the link to the UML code used to generate some of the figures in the present document.

Table A.4-1: Link to UML code

Figure	Link to UML code
A.1	https://forge.etsi.org/rep/set/etsi-ts-103-834-part-1/blob/17.0.0/figures/Figure%20A.1.plantuml
B.1	https://forge.etsi.org/rep/set/etsi-ts-103-834-part-1/blob/17.0.0/figures/Figure%20B.1.plantuml
C.1	https://forge.etsi.org/rep/set/etsi-ts-103-834-part-1/blob/17.0.0/figures/Figure%20C.1.plantuml
D.1	https://forge.etsi.org/rep/set/etsi-ts-103-834-part-1/blob/17.0.0/figures/Figure%20D.1.plantuml

Annex B (normative): TTI control service protocol

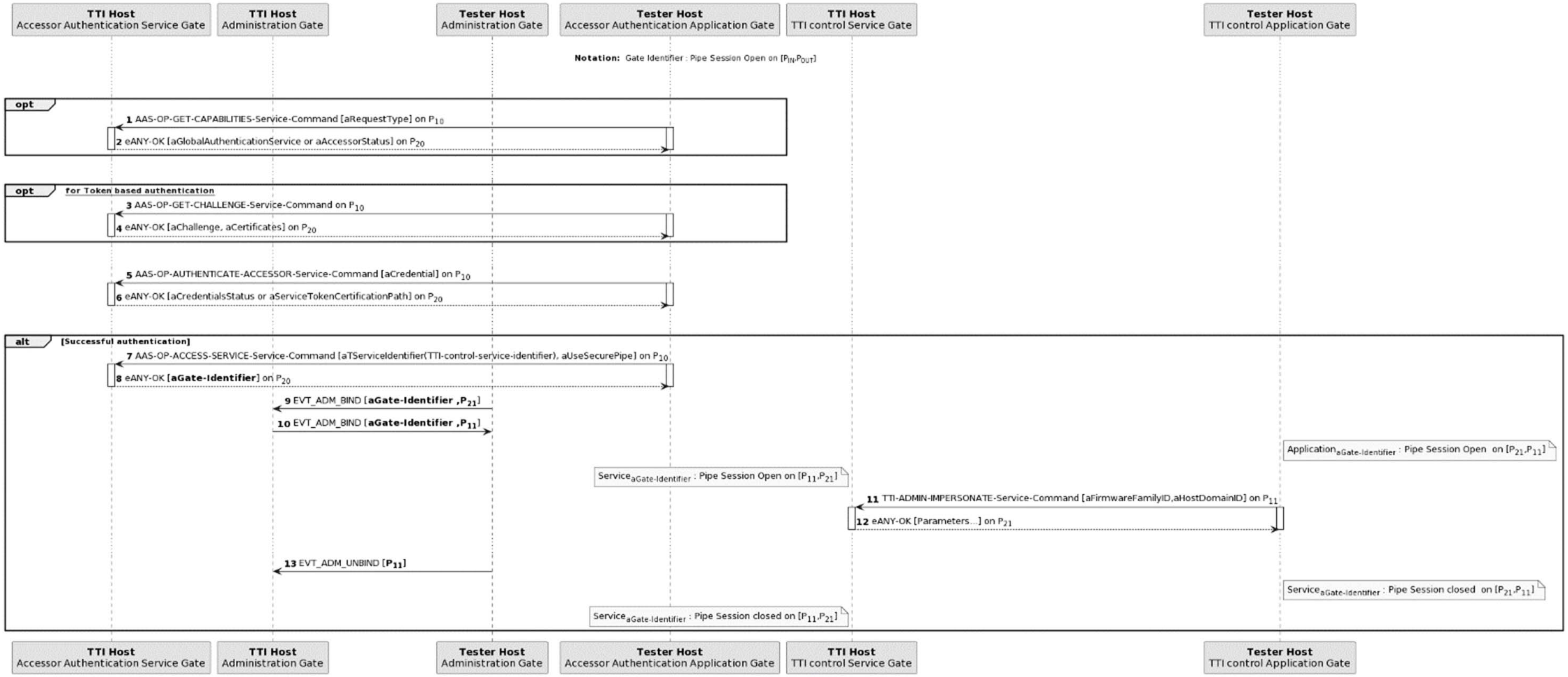
TTI Host Accessor Authentication Service Gate
TTI Host Administration Gate
Tester Host Administration Gate
Tester Host Accessor Authentication Application Gate
TTI Host TTI control Service Gate
Tester Host TTI control Application Gate
Notation: Gate Identifier : Pipe Session Open on [P_{IN},P_{OUT}]
opt
1 AAS-OP-GET-CAPABILITIES-Service-Command [aRequestType] on P_{10}
2 eANY-OK [aGlobalAuthenticationService or aAccessorStatus] on P_{20}
opt for Token based authentication
3 AAS-OP-GET-CHALLENGE-Service-Command on P_{10}
4 eANY-OK [aChallenge, aCertificates] on P_{20}
5 AAS-OP-AUTHENTICATE-ACCESSOR-Service-Command [aCredential] on P_{10}
6 eANY-OK [aCredentialsStatus or aServiceTokenCertificationPath] on P_{20}
alt [Successful authentication]
7 AAS-OP-ACCESS-SERVICE-Service-Command [aTServiceIdentifier(TTI-control-service-identifier), aUseSecurePipe] on P_{10}
8 eANY-OK [aGate-Identifier] on P_{20}
9 EVT_ADM_BIND [aGate-Identifier ,P_{21}]
10 EVT_ADM_BIND [aGate-Identifier ,P_{11}]
$\text{Application}_{\text{aGate-Identifier}}$: Pipe Session Open on [P_{21},P_{11}]
$\text{Service}_{\text{aGate-Identifier}}$: Pipe Session Open on [P_{11},P_{21}]
11 TTI-ADMIN-IMPERSONATE-Service-Command [aFirmwareFamilyID,aHostDomainID] on P_{11}
12 eANY-OK [Parameters...] on P_{21}
13 EVT_ADM_UNBIND [P_{11}]
$\text{Service}_{\text{aGate-Identifier}}$: Pipe Session closed on [P_{21},P_{11}]
$\text{Service}_{\text{aGate-Identifier}}$: Pipe Session closed on [P_{11},P_{21}]
TTI Host Accessor Authentication Service Gate
TTI Host Administration Gate
Tester Host Administration Gate
Tester Host Accessor Authentication Application Gate
TTI Host TTI control Service Gate
Tester Host TTI control Application Gate

Figure B.1: TTI control service session opening

The procedure has 13 steps:

1) The TT Host requests the authentication service capability from the TTI Host by sending the AAS-OP-GET-CAPABILITIES-Service-Command command.

2) The TTI Host returns the accessor authentication service capability.

3) The TT Host requests the authentication of an accessor by sending the command AAS-OP-GET-CHALLENGE-Service-Command command.

4) The TTI Host confirms the successful operation.

5) The TT Host requests the authentication of an accessor by sending the command AAS-OP-AUTHENTICATE-ACCESSOR-Service-Command command with its credentials.

6) The TTI Host confirms the successful authentication of the accessor.

7) The TT Host requests a session to a service by sending the command AAS-OP-ACCESS-SERVICE-Service-Command with the service identified by aServiceIdentifier containing the TTI control service identifier as defined in table 6.3.

8) The TTI Host dynamically creates a gate to the requested service and sends an answer to the TT Host with the identifier of the dynamically created gate.

9) The TT Host triggers the opening of a pipe session on the TTI control service gate.

10) The TTI Host confirms the pipe session.

11) The TT Host requests a host impersonation operation in using the TTI-ADMIN-IMPERSONATE-Service-Command command.

12) The TTI Host confirms the operation.

13) The TT Host closes the pipe session to the TTI control service gate.

Annex C (normative): TTI data service protocol

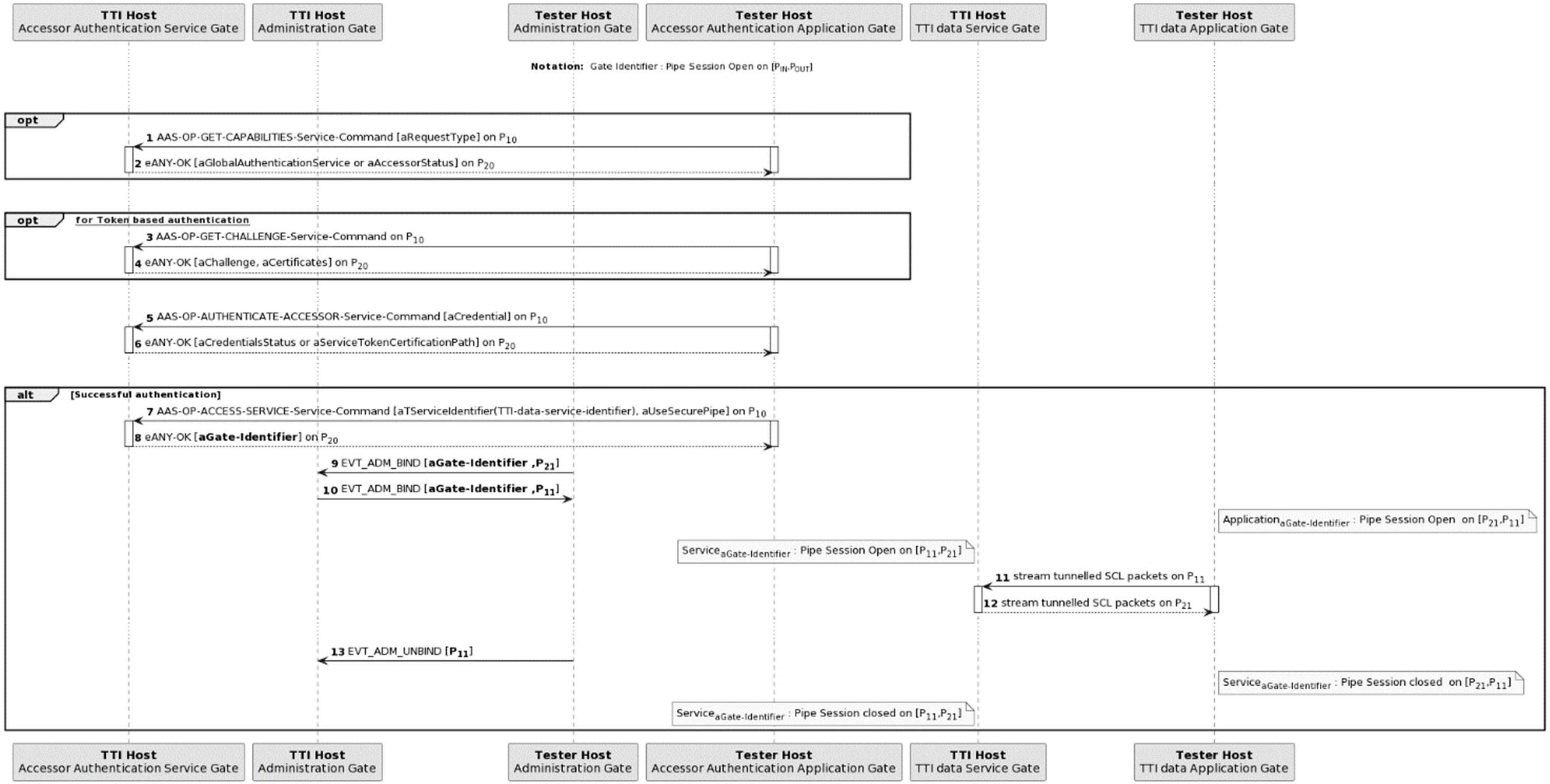
TTI Host Accessor Authentication Service Gate
TTI Host Administration Gate
Tester Host Administration Gate
Tester Host Accessor Authentication Application Gate
TTI Host TTI data Service Gate
Tester Host TTI data Application Gate
Notation: Gate Identifier : Pipe Session Open on [P_{IN},P_{OUT}]
opt
1 AAS-OP-GET-CAPABILITIES-Service-Command [aRequestType] on P_{10}
2 eANY-OK [aGlobalAuthenticationService or aAccessorStatus] on P_{20}
opt for Token based authentication
3 AAS-OP-GET-CHALLENGE-Service-Command on P_{10}
4 eANY-OK [aChallenge, aCertificates] on P_{20}
5 AAS-OP-AUTHENTICATE-ACCESSOR-Service-Command [aCredential] on P_{10}
6 eANY-OK [aCredentialsStatus or aServiceTokenCertificationPath] on P_{20}
alt [Successful authentication]
7 AAS-OP-ACCESS-SERVICE-Service-Command [aTServiceIdentifier(TTI-data-service-identifier), aUseSecurePipe] on P_{10}
8 eANY-OK [aGate-Identifier] on P_{20}
9 EVT_ADM_BIND [aGate-Identifier ,P_{21}]
10 EVT_ADM_BIND [aGate-Identifier ,P_{11}]
$\text{Application}_{\text{aGate-Identifier}}$: Pipe Session Open on [P_{21},P_{11}]
$\text{Service}_{\text{aGate-Identifier}}$: Pipe Session Open on [P_{11},P_{21}]
11 stream tunnelled SCL packets on P_{11}
12 stream tunnelled SCL packets on P_{21}
13 EVT_ADM_UNBIND [P_{11}]
$\text{Service}_{\text{aGate-Identifier}}$: Pipe Session closed on [P_{21},P_{11}]
$\text{Service}_{\text{aGate-Identifier}}$: Pipe Session closed on [P_{11},P_{21}]
TTI Host Accessor Authentication Service Gate
TTI Host Administration Gate
Tester Host Administration Gate
Tester Host Accessor Authentication Application Gate
TTI Host TTI data Service Gate
Tester Host TTI data Application Gate

Figure C.1: TTI data service session opening

The procedure has 13 steps:

1) The TT Host requests the authentication service capability from the TTI Host by sending the AAS-OP-GET-CAPABILITIES-Service-Command command.

2) The TTI Host returns the accessor authentication service capability.

3) The TT Host requests the authentication of an accessor by sending the command AAS-OP-GET-CHALLENGE-Service-Command command.

4) The TTI host confirms the successful operation.

5) The TT Host requests the authentication of an accessor by sending the command AAS-OP-AUTHENTICATE-ACCESSOR-Service-Command command with its credentials.

6) The TTI Host confirms the successful authentication of the accessor.

7) The TT Host requests a session to a service by sending the command AAS-OP-ACCESS-SERVICE-Service-Command with the service identified by aServiceIdentifier containing the TTI data service identifier as defined in table 6.3.

8) The TTI Host dynamically creates a gate to the requested service and sends an answer to the TT Host with the identifier of the dynamically created gate.

9) The TT Host triggers the opening of a pipe session on the TTI data service gate.

10) The TTI Host confirms the pipe session.

11) The TT Host sends tunnelled SCL packets as defined in clause 6.3.2.1.

12) The TTI Host receives tunnelled SCL packets as defined in clause 6.3.2.1.

13) The TT Host closes the pipe session to the TTI Control Service gate.

Annex D (normative): RDE data service protocol

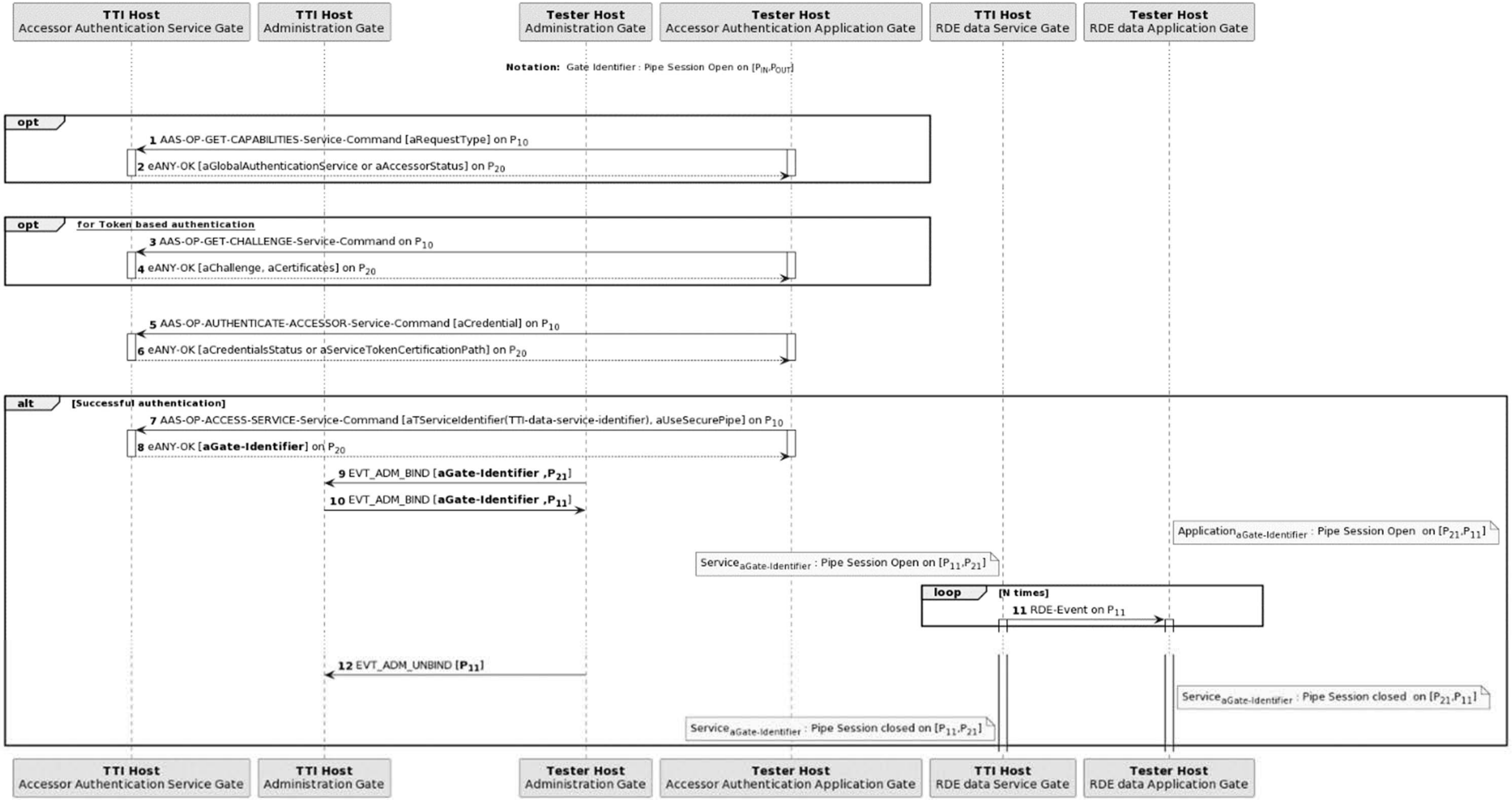
TTI Host
Accessor Authentication Service Gate
TTI Host
Administration Gate
Tester Host
Administration Gate
Tester Host
Accessor Authentication Application Gate
TTI Host
RDE data Service Gate
Tester Host
RDE data Application Gate
Notation: Gate Identifier : Pipe Session Open on [P_{IN},P_{OUT}]
opt
1 AAS-OP-GET-CAPABILITIES-Service-Command [aRequestType] on P_{10}
2 eANY-OK [aGlobalAuthenticationService or aAccessorStatus] on P_{20}
opt
for Token based authentication
3 AAS-OP-GET-CHALLENGE-Service-Command on P_{10}
4 eANY-OK [aChallenge, aCertificates] on P_{20}
5 AAS-OP-AUTHENTICATE-ACCESSOR-Service-Command [aCredential] on P_{10}
6 eANY-OK [aCredentialsStatus or aServiceTokenCertificationPath] on P_{20}
alt
[Successful authentication]
7 AAS-OP-ACCESS-SERVICE-Service-Command [aTServiceIdentifier(TTI-data-service-identifier), aUseSecurePipe] on P_{10}
8 eANY-OK [aGate-Identifier] on P_{20}
9 EVT_ADM_BIND [aGate-Identifier ,P_{21}]
10 EVT_ADM_BIND [aGate-Identifier ,P_{11}]
$\text{Application}_{\text{aGate-Identifier}}$: Pipe Session Open on [P_{21},P_{11}]
$\text{Service}_{\text{aGate-Identifier}}$: Pipe Session Open on [P_{11},P_{21}]
loop
[N times]
11 RDE-Event on P_{11}
12 EVT_ADM_UNBIND [P_{11}]
$\text{Service}_{\text{aGate-Identifier}}$: Pipe Session closed on [P_{21},P_{11}]
$\text{Service}_{\text{aGate-Identifier}}$: Pipe Session closed on [P_{11},P_{21}]
TTI Host
Accessor Authentication Service Gate
TTI Host
Administration Gate
Tester Host
Administration Gate
Tester Host
Accessor Authentication Application Gate
TTI Host
RDE data Service Gate
Tester Host
RDE data Application Gate

Figure D.1: RDE data service session opening

The procedure has 12 steps:

1) The TT Host requests the authentication service capability from the TTI host by sending the AAS-OP-GET-CAPABILITIES-Service-Command command.

2) The TTI Host returns the accessor authentication service capability.

3) The TT Host requests the authentication of an accessor by sending the command AAS-OP-GET-CHALLENGE-Service-Command command.

4) The TTI Host confirms the successful operation.

5) The TT Host requests the authentication of an accessor by sending the command AAS-OP-AUTHENTICATE-ACCESSOR-Service-Command command with its credentials.

6) The TTI Host confirms the successful authentication of the accessor.

7) The TT Host requests a session to a service by sending the command AAS-OP-ACCESS-SERVICE-Service-Command with the service identified by aServiceIdentifier containing the RDE data service identifier.

8) The TTI Host dynamically creates a gate to the requested service and sends an answer to the TT Host with the identifier of the dynamically created gate.

9) The TT Host triggers the opening of a pipe session on the TTI Control Service gate.

10) The TTI Host confirms the pipe session.

11) The TTI Host sends RDE-Event events until the pipe session closing.

12) The TT Host closes the pipe session to the TTI Control Service gate.

Annex E (normative): TTI_UL server and client connection procedure

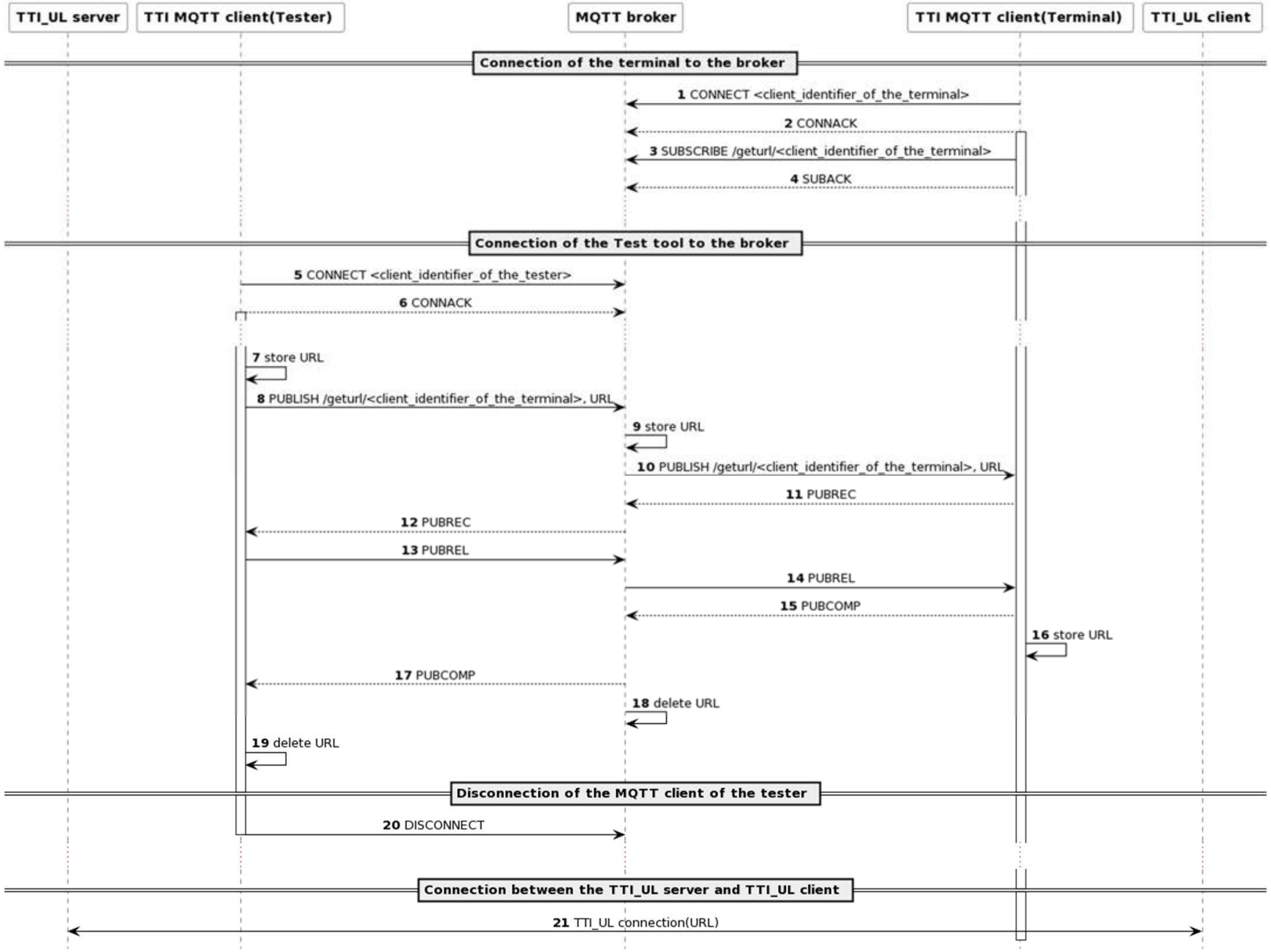
TTI_UL server
TTI MQTT client(Tester)
MQTT broker
TTI MQTT client(Terminal)
TTI_UL client
Connection of the terminal to the broker
1 CONNECT <client_identifier_of_the_terminal>
2 CONNACK
3 SUBSCRIBE /geturl/<client_identifier_of_the_terminal>
4 SUBACK
Connection of the Test tool to the broker
5 CONNECT <client_identifier_of_the_tester>
6 CONNACK
7 store URL
8 PUBLISH /geturl/<client_identifier_of_the_terminal>, URL
9 store URL
10 PUBLISH /geturl/<client_identifier_of_the_terminal>, URL
11 PUBREC
12 PUBREC
13 PUBREL
14 PUBREL
15 PUBCOMP
16 store URL
17 PUBCOMP
18 delete URL
19 delete URL
Disconnection of the MQTT client of the tester
20 DISCONNECT
Connection between the TTI_UL server and TTI_UL client
21 TTI_UL connection(URL)

Figure E.1: TTI client and server connection

The procedure supports basics of the MQTT specification for a publication with the QoS 2.

The procedure has 21 steps:

1) The TTI MQTT client of the terminal initiates a connection with a MQTT broker.

2) The broker accepts the connection.

3) The TTI MQTT client of the terminal subscribes to the topic /geturl/client_identifier_of_the_terminal.

4) The broker accepts the subscription.

5) The TTI MQTT client of the TT initiates a connection with a MQTT broker.

6) The broker accepts the connection.

7) The TTI MQTT client of the TT stores the URL.

8) The TTI MQTT client of the TT publishes its URL for given TTI MQTT client to the broker.

9) The broker stores the URL of the TTI_UL server

10) The broker forwards the publication to the TTI MQTT client of the terminal.

11) The TTI MQTT client of the terminal acknowledges the reception of the publication.

12) The broker forwards the acknowledge the reception of the publication.

13) The TTI MQTT client of the TT acknowledges the reception of the PUBREC message.

14) The broker forwards the acknowledge the reception of the PUBREC message.

15) The TTI MQTT client of the terminal confirms the completion of the transaction to the broker.

16) The TTI MQTT client of the terminal stores the URL of the TTI_UL server.

17) The broker forwards the completion of the transaction to TTI MQTT client of the TT.

18) The broker deletes the URL of the TTI_UL server

19) The TTI MQTT client of the TT may delete the URL of the TTI_UL server.

20) The TTI MQTT client of the TT disconnect from the broker.

21) The TTI_UL client connects the TTI_UL server by using its URL.

Annex F (informative): ASN.1 definition

F.1 End of ASN.1

```
-- ASN1START

END

-- ASN1STOP
```

Annex G (informative): List of IP binding standards

This clause defines the list of standards supporting the transport of IP packets for a given technology (IP binding). The table defines the list of recommended IP binding standards or specifications.

Table G.1: IP binding standards/specifications

Technology	IP binding standards
Bluetooth® Low Energy version 4 and above	IETF RFC 7668 [i.4] IPv6 over BLUETOOTH® Low Energy
WIFI	Native
GSM broadband	Native
USB	Remote Network Driver Interface Specification (RNDIS) Microsoft® [i.5] USB communications device class (usb.org). CDC EEM [i.6]

NOTE: The TT and the terminal may not support the same technology for conveying IP packets over a given protocol stack. In consequence, the direct connection between the TT and the terminal hosting the SUT is not mandatory. Any set of adapters may bridge the TT and the terminal to allow the indirect transport of IP packets between endpoint entities.

Annex H (informative): Change History

The table below indicates all changes that have been incorporated into the present document since it was placed under change control.

Change history								
Date	Meeting	Plenary Doc	CR	Rev	Cat	Subject/Comment	Old	New
08/12/2022	SET#108	SET(22)000230	-	-	-	Version 17.0.0 first publication	-	17.0.0

History

Document history		
V17.0.0	December 2022	Publication