ETSI TS 133 122 V18.4.0 (2024-07)

LTE;
5G;
Security aspects of Common API Framework (CAPIF)
for 3GPP northbound APIs
(3GPP TS 33.122 version 18.4.0 Release 18)

Reference
RTS/TSGS-0333122vi40

Keywords
5G,LTE

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

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.

Legal Notice

This Technical Specification (TS) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities. These shall be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between 3GPP and ETSI identities can be found under https://webapp.etsi.org/key/queryform.asp.

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.

Contents

Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

- x the first digit:
  - 1 presented to TSG for information;
  - 2 presented to TSG for approval;
  - 3 or greater indicates TSG 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.

1 Scope

The present document specifies the security architecture i.e., the security features and the security mechanisms for the common API framework (CAPIF) as per the architecture and procedures defined in 3GPP TS 23.222 [3].

2 References

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

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TS 33.310: "Network Domain Security (NDS); Authentication Framework (AF)".

[3] 3GPP TS 23.222: "Common API Framework for 3GPP Northbound APIs".

[4] IETF RFC 6749: "The OAuth 2.0 Authorization Framework".

[5] IETF RFC 6750: "The OAuth 2.0 Authorization Framework: Bearer Token Usage".

[6] IETF RFC 7519: "JSON Web Token (JWT)".

[7] IETF RFC 7515: "JSON Web Signature (JWS)".

[8] 3GPP TS 33.220: "Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (GBA)".

[9] Void

[10] 3GPP TS 33.210: "3G security; Network Domain Security (NDS); IP network layer security".

[11] IETF RFC 7636: " Proof Key for Code Exchange by OAuth Public Clients".

3 Definitions, symbols and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in 3GPP TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in 3GPP TR 21.905 [1].

3.2 Symbols

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

AEF_{PSK} Pre-Shared Key for AEF

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in 3GPP TR 21.905 [1].

AEF	API Exposing Function
API	Application Programming Interface
CAPIF	Common API Framework
JSON	JavaScript Object Notation
JWT	JSON Web Token
KDF	Key Derivation Function
PKI	Public Key Infrastructure
PSK	Pre-Shared Key
RNAA	Resource owner-aware northbound API access
TLS	Transport Layer Security

4 Security requirements

4.1 General

Architectural requirements pertaining to CAPIF security are found in 3GPP TS 23.222 [3]. The following are CAPIF derived security requirements.

4.2 Common security requirements

Security requirements that are applicable to all CAPIF entities are:

- [CAPIF-SEC-4.2-a] The CAPIF shall provide mechanisms to hide the topology of the PLMN trust domain from the API invokers accessing the service APIs from outside the PLMN trust domain.
- [CAPIF-SEC-4.2-b] The CAPIF shall provide mechanisms to hide the topology of the 3rd party API provider trust domain from the API invokers accessing the service APIs from outside the 3rd party API provider trust domain.
- [CAPIF-SEC-4.2-c] The CAPIF shall provide authorization mechanism for service APIs from the 3rd party API providers.
- [CAPIF-SEC-4.2-d] The CAPIF shall support a common security mechanism for all API implementations to provide confidentiality and integrity protection.
- [CAPIF-SEC-4.2-e] API invoker authentication and authorization shall support all deployment models listed in 3GPP TS 23.222 [3].
- [CAPIF-SEC-4.2-f] The API invoker and CAPIF should enforce the result of the authentication for the duration of communications (e.g. by integrity protection or implicit authentication by encryption with a key that is derived from the authentication and is unknown to the adversary).

4.3 Security requirements on the CAPIF-1/1e reference points

The CAPIF-1/1e reference points between the API invoker and the CAPIF core function shall fulfil the following requirements:

- [CAPIF-SEC-4.3-a] Mutual authentication between the API invoker and the CAPIF Core function shall be supported.
- [CAPIF-SEC-4.3-b] The transport of messages over the CAPIF-1 and CAPIF-1e reference points shall be integrity protected.

- [CAPIF-SEC-4.3-c] The transport of messages over the CAPIF-1 and CAPIF-1e reference points shall be protected from replay attacks.
- [CAPIF-SEC-4.3-d] The transport of messages over the CAPIF-1 and CAPIF-1e reference points shall be confidentiality protected.
- [CAPIF-SEC-4.3-e] Privacy of the 3GPP user over the CAPIF-1 and CAPIF-1e reference points shall be protected.
- [CAPIF-SEC-4.3-f] The CAPIF core function shall authorize the API invoker prior to the API invoker accessing the AEF.
- [CAPIF-SEC-4.3-g] The CAPIF core function shall authorize the API invoker prior to accessing the discover service API.
- [CAPIF-SEC-4.3-h] The CAPIF core function shall authenticate the API invoker's onboarding request.
- [CAPIF-SEC-4.3-i] The CAPIF core function shall authenticate the API invoker's offboarding request.

4.4 Security requirements on the CAPIF-2/2e reference points

The CAPIF-2/2e reference points between the API invoker and API exposing function shall fulfil the following requirements:

- [CAPIF-SEC-4.4-a] Mutual authentication between the API invoker and the API exposing function shall be supported.
- [CAPIF-SEC-4.4-b] The transport of messages over the CAPIF-2 and CAPIF-2e reference points shall be integrity protected.
- [CAPIF-SEC-4.4-c] The transport of messages over the CAPIF-2 and CAPIF-2e reference points shall be protected from replay attacks.
- [CAPIF-SEC-4.4-d] The transport of messages over the CAPIF-2 and CAPIF-2e reference points shall be confidentiality protected.
- [CAPIF-SEC-4.4-e] Privacy of the 3GPP user over the CAPIF-2 and CAPIF-2e reference points shall be protected.
- [CAPIF-SEC-4.4-f] The API exposing function shall determine whether API invoker is authorized to access service API.

4.5 Security requirements on the CAPIF-3/4/5 reference points

The security requirements for CAPIF-3/4/5 reference points are:

- [CAPIF-SEC-4.5-a] The transport of messages over the CAPIF-3/4/5 reference points shall be integrity protected.
- [CAPIF-SEC-4.5-b] The transport of messages over the CAPIF-3/4/5 reference points shall be confidentiality protected.
- [CAPIF-SEC-4.5-c] The transport of messages over the CAPIF-3/4/5 reference points shall be protected from replay attacks.
- [CAPIF-SEC-4.5-d] The CAPIF core function shall be able to authenticate the service API publishers to publish and manage the service API information.
- [CAPIF-SEC-4.5-e] The CAPIF core function shall be able to authorize the service API publishers to publish and manage the service API information.
- [CAPIF-SEC-4.5-f] The CAPIF core function shall be able to request explicit grant of new API invoker’s onboarding.

- [CAPIF-SEC-4.5-g] The CAPIF core function shall be able to authenticate the API Management function's registration request for API Provider domain functions.
- [CAPIF-SEC-4.5-h] The CAPIF core function shall be able to authenticate the API Management function's registration update request for API provider domain functions.

4.6 Security requirements on the CAPIF-3e/4e/5e reference points

The security requirements for CAPIF-3e/4e/5e reference points are:

- [CAPIF-SEC-4.6 -a] The transport of messages over the CAPIF-3e/4e/5e reference points shall be integrity protected.
- [CAPIF-SEC-4. 6 -b] The transport of messages over the CAPIF-3e/4e/5e reference points shall be confidentiality protected.
- [CAPIF-SEC-4. 6 -c] The transport of messages over the CAPIF-3e/4e/5e reference points shall be protected from replay attacks.
- [CAPIF-SEC-4. 6 -d] The CAPIF core function shall be able to authenticate the service API publishers to publish and manage the service API information.
- [CAPIF-SEC-4. 6 -e] The CAPIF core function shall be able to authorize the service API publishers to publish and manage the service API information.
- [CAPIF-SEC-4. 6 -f] The CAPIF core function shall be able to request explicit grant of new API invoker's onboarding.
- [CAPIF-SEC-4.6-g] The CAPIF core function shall be able to authenticate the API Management function's registration request for API Provider domain functions.
- [CAPIF-SEC-4.6-h] The CAPIF core function shall be able to authenticate the API Management function's registration update request for API provider domain functions.

4.7 Security requirements on the CAPIF-7/7e reference points

The security requirements for CAPIF-7/7e reference points are:

- [CAPIF-SEC-4.7-a] The transport of messages over the CAPIF-7 and CAPIF-7e reference points shall be integrity protected.
- [CAPIF-SEC-4.7-b] The transport of messages over the CAPIF-7 and CAPIF-7e reference points shall be protected from replay attacks.
- [CAPIF-SEC-4.7-c] The transport of messages over the CAPIF-7 and CAPIF-7e reference points shall be confidentiality protected.
- [CAPIF-SEC-4.7-d] Privacy of the 3GPP user over the CAPIF-7 and CAPIF-7e reference points shall be protected.
- [CAPIF-SEC-4.7-e] The API exposing function (destination AEF handling service API) shall determine whether AEF that is topology hiding entity, is authorized to access service API.

4.8 Security requirements on the CAPIF-8reference points

CAPIF-8 interface is not in scope of the present document. Nevertheless, integrity and confidentiality protection, protection against replay attacks, privacy of the resource owner, authentication between the resource owner and the CCF need to be addressed by mechanism(s) which are out of 3GPP scope.

5 Functional security model

5.1 General functional security model

Figure 5.1-1 shows the functional security model for the CAPIF architecture. The interfaces CAPIF-1, CAPIF-1e, CAPIF-2, CAPIF-2e, CAPIF-3, CAPIF-4, CAPIF-5, CAPIF-3e, CAPIF-4e, CAPIF-5e, CAPIF-7 and CAPIF-7e are defined in 3GPP TS 23.222 [3] and support the CAPIF functionality defined in 3GPP TS 23.222 [3]. CAPIF-1, CAPIF-2, CAPIF-3, CAPIF-4, CAPIF-5 and CAPIF-7 are interfaces that lie within the PLMN trust domain while the CAPIF-1e, CAPIF-2e, CAPIF-3e, CAPIF-4e, CAPIF-5e and CAPIF-7e interfaces are CAPIF core and AEF access points for API Invokers outside of the PLMN trust domain.

Security for the CAPIF-1, CAPIF-2, CAPIF-3, CAPIF-4, CAPIF-5 and CAPIF-7 interfaces support TLS and are defined in subclauses 6.2, 6.4 and 6.6 of the present document. Security for the CAPIF-1e, CAPIF-2e and CAPIF-7e interfaces support TLS, and are defined in subclause 6.3, subclause 6.5, and subclause 6.9 respectively.

Security for the CAPIF-3e, CAPIF-4e and CAPIF-5e interfaces support NDS/IP security to secure communication between different IP security domains. This avoids multiple secure connections between API provider domain and CAPIF core domain by leveraging the NDS/IP security procedures specified in TS 33.210 [2].

Authentication and authorization are required for both API invokers that lie within the PLMN trust domain and API invokers that lie outside of the PLMN trust domain. For an API invoker that is outside of the PLMN trust domain, the CAPIF core function in coordination with the API exposing function utilizes the CAPIF-1e, CAPIF-2e and the CAPIF-3 interfaces to onboard, authenticate and authorize the API invoker prior to granting access to CAPIF services. Security flow diagrams for onboarding security, CAPIF-1e security and CAPIF-2e security can be found in Annex B. When the API invoker is within the PLMN trust domain, the CAPIF core function in coordination with the API exposing function perform authentication and authorization of the API invoker via the CAPIF-1, the CAPIF-2 and the CAPIF-3 interfaces prior to granting access to CAPIF services. Authentication and authorization of API invokers (both internal and external to the PLMN trust domain) is specified in clause 6 of the present document.

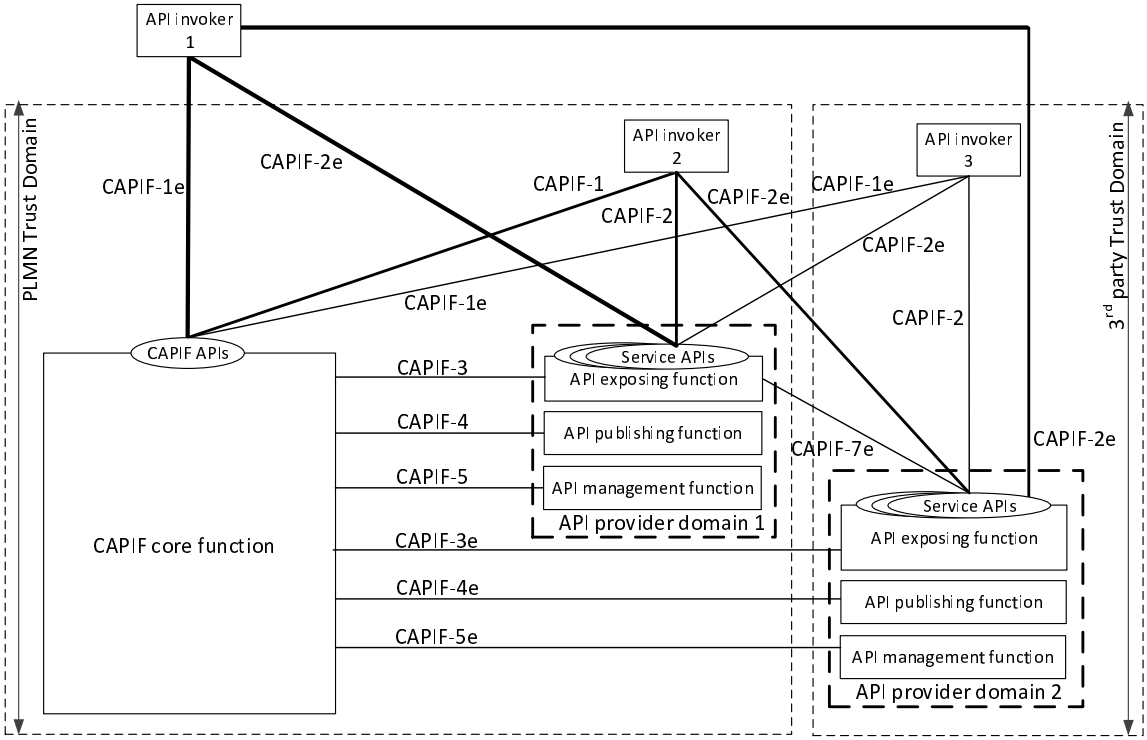

Figure 5-1: CAPIF functional security model

5.2 Functional security model supporting RNAA

Figure 5.2-1 shows the functional security architecture of CAPIF framework when RNAA is supported. The resource owner can be the user of the UE or the owner of the subscription depending on the use case and regulations.

The resource owner function (ROF) may be deployed on the UE.

The authorization function is a part of the CCF.

The API invoker is the OAuth 2.0 client.

The OAuth 2.0 client and the CCF shall communicate using https.

Different functional security models can be envisioned for API invoker in relation to the ROF:

- API invoker can be part of the UE and located on the device;
- API invoker can be independent from the UE but still located on the device (e.g., deployed by a third party);
- API invoker can be independent from the UE and located outside of the device (e.g., a game server).

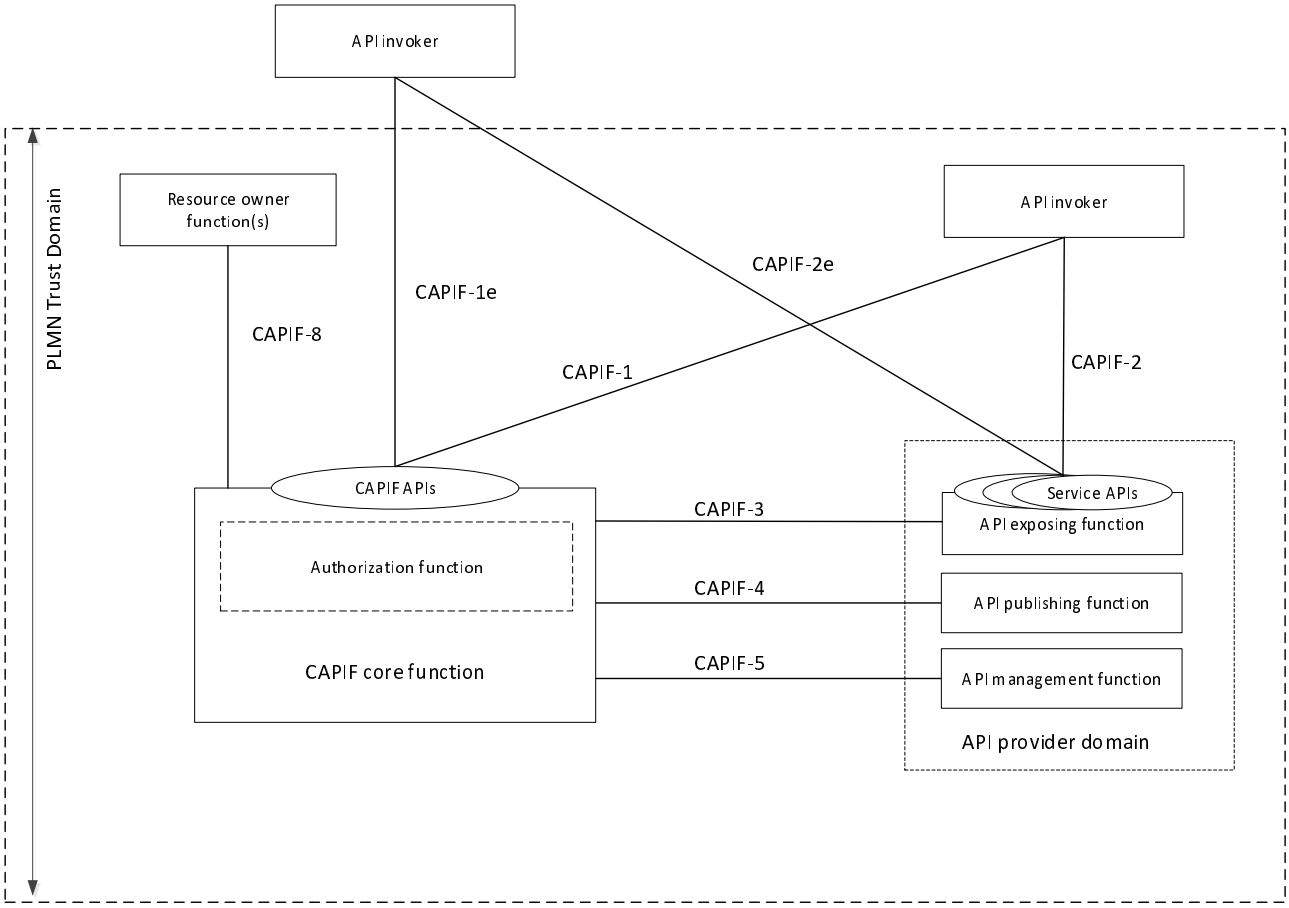

Figure 5.2-1: CAPIF supporting RNAA functional security model

6 Security procedures

6.1 Security procedures for API invoker onboarding

The API invoker and the CAPIF core function shall follow the procedure in this subclause to secure and authenticate the onboarding of the API invoker to the CAPIF core function. The API invoker and the CAPIF core function shall establish a secure session using TLS. Security profiles for TLS implementation and usage shall follow the provisions given in TS 33.310 [2], Annex E .

With a secure session established, the API Invoker sends an Onboard API Invoker Request message to the CAPIF core function. The Onboard API Invoker Request message carries an onboard credential obtained during pre-provisioning of the onboard enrolment information, which may be an OAuth 2.0 [4] access token. When the OAuth 2.0 token based mechanism is used as the onboarding credential, the access token shall be encoded as JSON web token as specified in IETF RFC 7519 [6], shall include the JSON web signature as specified in IETF RFC 7515 [7], and shall be validated per OAuth 2.0 [4], IETF RFC 7519 [6] and IETF RFC 7515 [7]. Other credentials may also be used (e.g. message digest).

Figure 6.1-1 details the security information flow for the API invoker onboarding procedure. The OAuth 2.0 token based authentication credential is shown in this example.

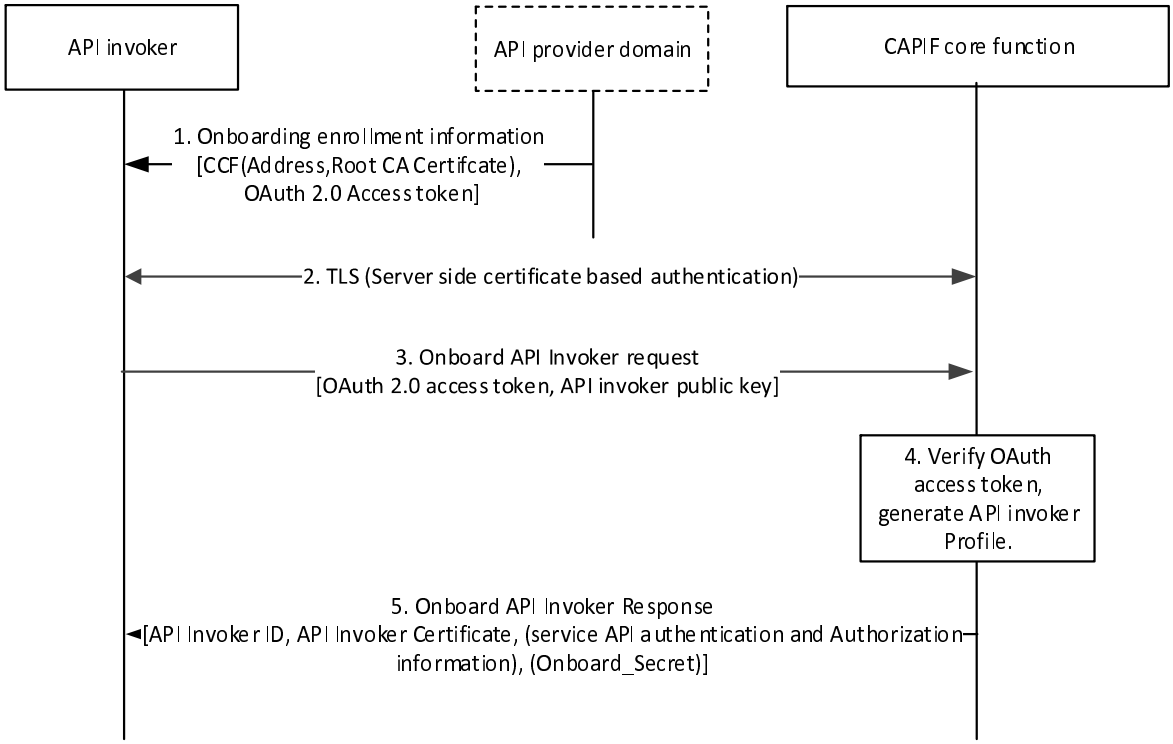

Figure 6.1-1: Security procedure for API invoker onboarding

1. As a prerequisite to the onboarding procedure, the API invoker obtains onboarding enrolment information from the API provider domain. The onboarding enrolment information is used to authenticate and establish a secure TLS communication with the CAPIF core function during the onboarding process. The enrolment information includes details of the CAPIF core function (Address, and Root CA certificate) and includes an onboarding credential (the OAuth 2.0 [4] access token).

NOTE 1: The procedure used to obtain the enrolment information by the API invoker is out of scope of the present document.

2. The API invoker and CAPIF core function shall establish a secure session based on TLS (Server side certificate authentication). The API invoker shall use the enrolment information obtained in step 1 to establish the TLS session with the CAPIF core function.

3. After successful establishment of the TLS session, the API invoker shall send an Onboard API invoker request message to the CAPIF core function along with the enrolment credential (OAuth 2.0 [4] access token). The API invoker generates the key pair {Private Key, Public key} and provides the public key along with the Onboard API invoker request.

4. The CAPIF core function shall validate the enrolment credential (OAuth 2.0 [4] access token). If validation of the credential (the OAuth 2.0 [4] access token in this example) is successful, the CAPIF core function shall generate an API invoker's profile as specified in TS 23.222 [3] which may contain the selected method for AEF authentication and authorization between the API Invoker and the AEF (see subclause 6.5.2). The CAPIF core function may generate API invoker's certificate on its own, for the assigned API invoker identity and public key. This certificate shall be used by the API invoker for subsequent authentication procedures with the CAPIF core function and may be used for establishing a secure connection and authentication with the API Exposing Function. The CAPIF core function may optionally generate an Onboard_Secret if the subscribed Service API uses Method 3 (as specified in clause 6.5.2.3 of the present document) for CAPIF-2e security. The Onboard_Secret value remains the same during the lifetime of the onboarding, and shall be bound to the CAPIF core function specific API Invoker ID.

NOTE 2: When API invoker's client certificate is issued by the third party, then in Step 3 the API invoker can additionally include the certificate in Onboard API Invoker request message. If the CAPIF core function trusts the issuer of the API invoker's client certificate, then the CAPIF Core Function includes the provided certificate in the API invoker's profile, in step 4. It is up to the CAPIF domain policy to accept the client certificates issued by third party.

5. The CAPIF core function shall respond with an Onboard API invoker response message. The response shall include the CAPIF core function assigned API invoker ID, AEF Authentication and authorization information (if generated in step 4), API invoker's certificate and the API invoker Onboard_Secret (if generated by the CAPIF core function).

6.2 Security procedures for CAPIF-1 reference point

TLS shall be used to provide integrity protection, replay protection and confidentiality protection. The support of TLS is mandatory and optional to use based on the domain administrator's policy to protect interfaces within the trusted domain.

The procedure in subclause 6.3 of the present document shall be followed unless the security of CAPIF-1 reference point is provided by other means.

6.3 Security procedures for CAPIF-1e reference point

6.3.1 Authentication and authorization

6.3.1.1 General

For authentication of the CAPIF-1e reference point, mutual authentication based on client and server certificates shall be performed between the CAPIF core function and the API invoker, using TLS.

Certificate based authentication shall follow the profiles given in 3GPP TS 33.310 [2], subclauses 6.1.3a and 6.1.4a. The structure of the PKI used for the certificate is out of scope of the present document.

TLS [9] shall be used to provide integrity protection, replay protection and confidentiality protection for CAPIF-1e interface. The support of TLS on CAPIF-1e interface is mandatory. Security profiles for TLS implementation and usage shall follow the provisions given in TS 33.310 [2], Annex E.

6.3.1.2 Security method negotiation

The API invoker and the CAPIF core function shall negotiate a security method that shall be used by the API invoker and the API exposing function for CAPIF-2e interface authentication and protection. After successful mutual authentication on CAPIF-1e interface, based on the API invoker's subscribed service APIs, access scenarios (whether the API invoker access the AEF prior to service API invocation or upon the service API invocation) and AEF capabilities, the CAPIF core function shall choose the security method and sends the chosen security methods along with the information required for authentication of the API invoker at the AEF to the API invoker. The information may include the validity time of the CAPIF-2e credentials. This is depicted in figure 6.3.1-1.

Pre-conditions:

1. The API invoker is onboarded with the CAPIF core function.

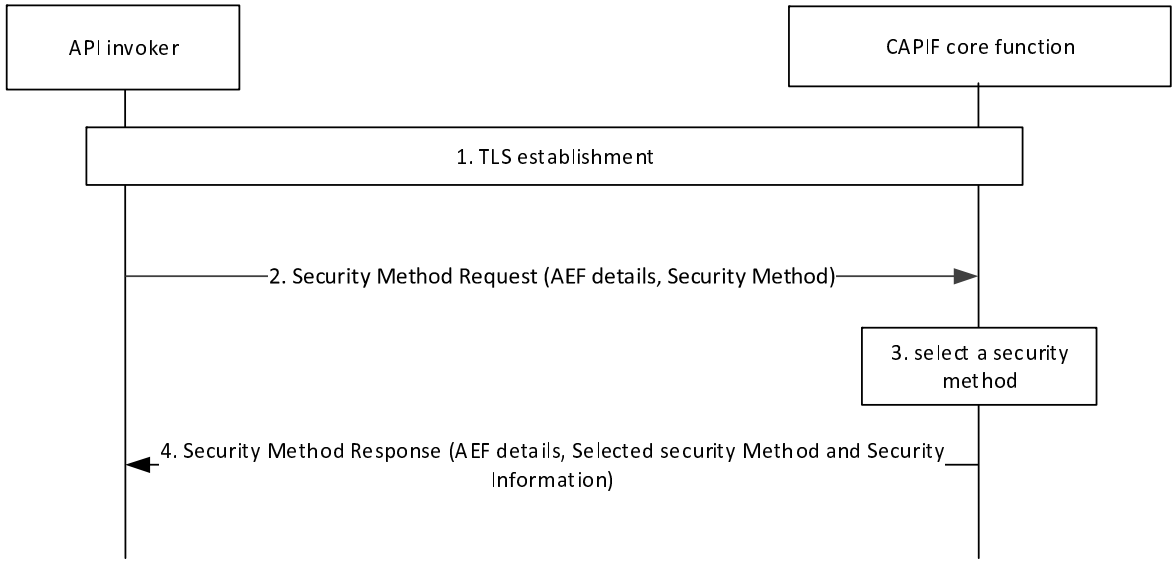

Figure 6.3.1-1: Selection of security method to be used in CAPIF-2/2e reference point

1. Mutual authentication based on client and server certificates shall be established using TLS between the API invoker and the CAPIF core function. The client certificate that was provided to the API invoker as the result of successful onboarding is used based on the description in subclause 6.1 of the present document.

2. The API invoker may send CAPIF-2/2e security capability information to the CAPIF core function in the Security Method Request message, indicating the list of security methods that the API invoker supports over CAPIF-2/2e reference point for each AEF.

3. The CAPIF core function shall select a security method to be used over CAPIF-2/2e reference point for each requested AEF, taking into account the information from the API invoker in step 2, access scenarios and AEF capabilities.

4. The CAPIF core function shall send Security Method Response message to the API invoker, indicating the selected security method for each AEF, any security information related to the security method. The API invoker shall use this method in the subsequent communication establishment with the API exposing function over CAPIF-2/2e reference point, as described in subclause 6.5 of the present document.

6.3.1.3 API discovery

After successful authentication between API invoker and CAPIF core function, the CAPIF core function shall decide whether the API invoker is authorized to perform discovery based on API invoker ID and discovery policy.

6.3.1.4 Topology hiding

When topology hiding is enabled, the CAPIF core function shall respond to service APIs discovery requests with AEF information, which exposes the service API and acts as topology hiding entity.

6.4 Security procedures for CAPIF-2 reference point

TLS shall be used to provide integrity protection, replay protection and confidentiality protection. The support of TLS is mandatory and optional to use based on the domain administrator's policy to protect interfaces within the trusted domain.

The procedure in subclause 6.5 of the present document shall be followed unless the security of CAPIF-2 reference point is provided by other means.

If the domain administrator's policy to authorize the API invoker's service API invocation requests is set, the API invoker's authorization shall be performed according to the authorization mechanisms specified for CAPIF-2e reference point in subclause 6.5 of the present document.

6.5 Security procedures for CAPIF-2e reference point

6.5.1 General

Based on the selected security method by the CAPIF Core Function (c.f., subclause 6.3.1), one of the methods specified in subclause 6.5.2 shall be used between the API invoker and a 3GPP defined API exposing function for CAPIF-2e interface authentication and protection.

NOTE: The security methods for the AEFs not defined by 3GPP are out of scope of the present document.

6.5.2 Authentication and authorization

6.5.2.1 Method 1 – Using TLS-PSK

The API invoker and the API exposing function shall follow the procedure in this sub-clause to establish dedicated secure session using TLS connection based on Pre-Shared Key (PSK). CAPIF-1e authentication shall be used to bootstrap a Pre-Shared key for authenticating a TLS connection for CAPIF-2e. It is assumed that both the API invoker and the CAPIF core function are pre-provisioned with certificates. The TLS profile as specified in Annex E of TS 33.310 [2] shall be used.

Figure 6.5.2.1-1 details the message flow between the API invoker, the CAPIF core function and the API exposing function, to establish secure CAPIF-2e interface using a pre-shared key for authentication.

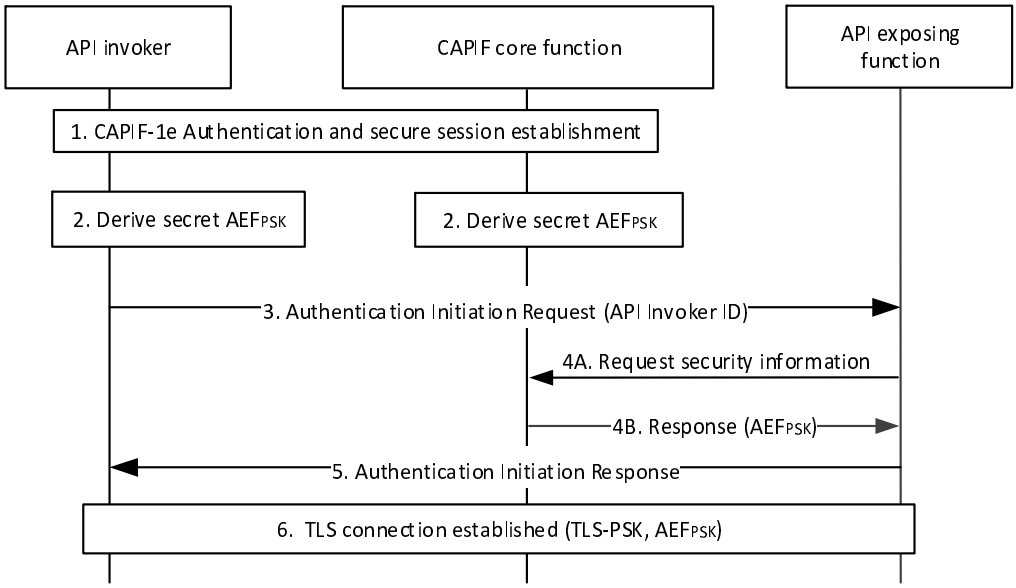

Figure 6.5.2.1-1: CAPIF-2e interface authentication and protection using TLS-PSK

1. CAPIF-1e authentication and secure session is established as specified in subclause 6.3.1 of the present document. The CAPIF core function shall provide the validity timer value for the key $AEF_{PSK.}$

2. After successful establishment of TLS on CAPIF-1e, the API invoker and the CAPIF core function shall derive the key AEF_{PSK}.

   The Key AEF_{PSK} shall be bound to an AEF and shall be derived as specified in Annex A. The API invoker and the CAPIF core function starts the validity timer for the key AEF_{PSK}.

3. The API Invoker shall send Authentication Initiation Request to the AEF, including the CAPIF core function assigned API invoker ID.

   Steps 1 and 2 of this procedure may be skipped if the API invoker is already in possession of a valid key AEF_{PSK}. In this case, the API invoker begins the procedure at step 3.

NOTE: Void.

4. The AEF shall request for security information from the CAPIF Core Function to perform authentication and secure interface establishment with the API invoker, if the AEF does not have a valid key. The CAPIF Core Function provides the security information related to the chosen security method (TLS-PSK: AEF_{PSK}) to the AEF over CAPIF-3 reference point. The CAPIF core function shall provide the remaining validity timer value for the key AEF_{PSK}.

5. After fetching the relevant security information (AEF_{PSK}) for the authentication, the AEF shall send Authentication Initiation Response message to API invoker to initiate the TLS session establishment. The AEF starts the validity timer based on the value received from the CAPIF core function in step 4.

6. The API Invoker and the AEF shall perform mutual authentication using the key AEF_{PSK} and establish TLS session over the CAPIF-2e.

After successful establishment of TLS on CAPIF-2e reference point, the API exposing function shall authorize the API invoker's service API invocation request based on authorization information obtained from CAPIF core function as specified in subclause 8.16 of TS 23.222 [3].

6.5.2.2 Method 2 – Using PKI

The API invoker and the API exposing function shall follow the procedure in this subclause to establish dedicated secure session over CAPIF-2e using TLS based on certificate based mutual authentication. It is assumed that both API invoker and API exposing function are pre-provisioned with certificates.

Figure 6.5.2.2-1 details the message flow between the API invoker, the CAPIF core function and the API exposing function related to this security method.

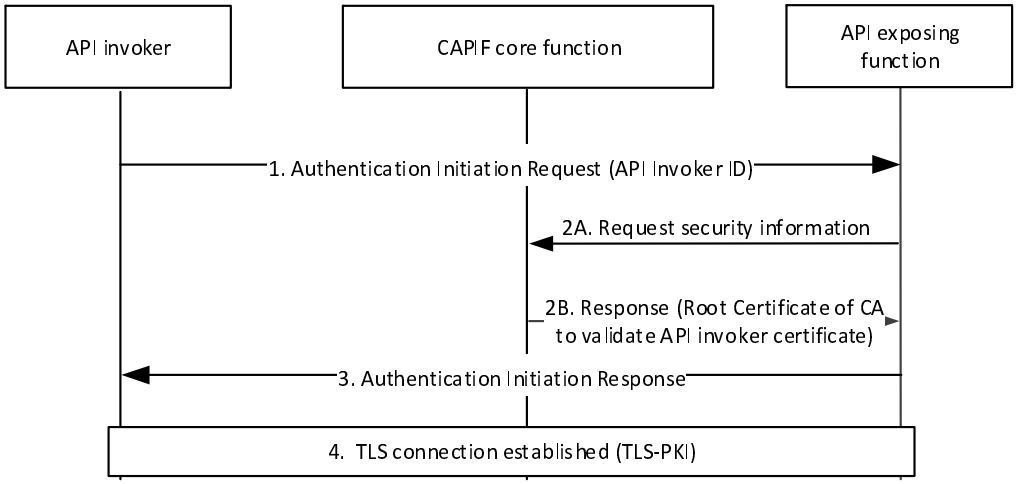

Figure 6.5.2.2-1: CAPIF-2e interface authentication and protection using certificate based mutual authentication

1. The API invoker shall send Authentication Initiation Request to the AEF, including API invoker ID.

2. The AEF shall request for security information from the CAPIF Core Function to perform authentication and secure interface establishment with the API invoker. The CAPIF Core Function provides the security information related to the chosen security method (TLS-PKI) to the AEF over CAPIF-3 reference point. CAPIF core function may return API invoker's root CA certificate for the AEF to validate the API invoker's certificate.

3. After fetching the relevant security information for the authentication, AEF shall send Authentication Initiation Response message to API invoker to initiate the TLS session establishment procedure.

4. Then the API Invoker and the AEF shall perform mutual authentication using certificates and establish TLS session over the CAPIF-2e. Certificate based authentication shall follow the profiles given in 3GPP TS 33.310 [2], clauses 6.1.3a and 6.1.4a. The structure of the PKI used for the certificate is out of scope of the present document.

After successful establishment of TLS on CAPIF-2e reference point, the API exposing function shall authorize the API invoker's service API invocation request based on authorization information obtained from CAPIF core function as specified in subclause 8.16 of TS 23.222 [3].

6.5.2.3 Method 3 – TLS with OAuth token

This method details establishment of secure channel over CAPIF-1e, CAPIF-2e reference points, and uses the OAuth 2.0 [4] token based mechanism to authorize and honour API invoker's northbound API invocations to the API exposing function. Figure 6.5.2.3-1 details security information flows between the API invoker, the CAPIF core function and the API exposing function. It is assumed that the API invoker, the CAPIF core function and the AEF are pre-provisioned with the appropriate credentials and related information to establish a secure session.

As per OAuth 2.0 [4], the CAPIF core function shall perform the functionality of the authorization server and provide the token endpoint, the API invoker shall perform the function of the client functionality, while the API exposing function shall perform the resource server functions. The API invoker client (client endpoint) shall be registered as a

confidential client type with an authorization grant type of 'client credentials'. The authorization shall be previously arranged in the CAPIF core function. The access token shall follow the profile described in annex C.

NOTE: How the authorization is pre-arranged (pre-configured) with the CAPIF core function is out of scope of the present document

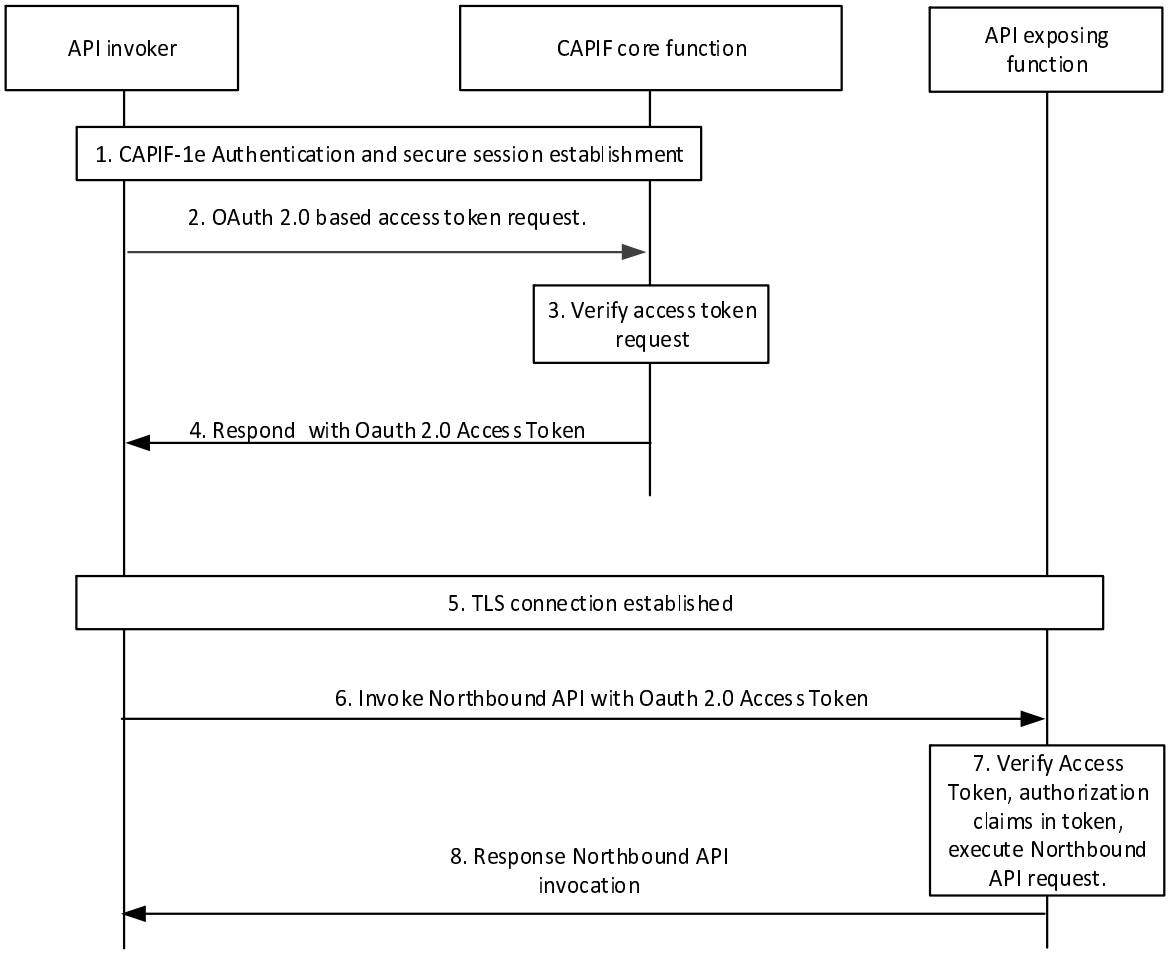

Figure 6.5.2.3-1: CAPIF-2e interface authentication and protection using Access Tokens

1. CAPIF-1e authentication and secure session establishment is performed as specified in subclause 6.3.1.

2. After successful establishment of TLS session over CAPIF-1e, as described in subclause 6.3.1 of the present document, the API invoker shall send an Access Token Request message to the CAPIF core function as per the OAuth 2.0 [4] specification.

3. The CAPIF core function shall verify the Access Token Request message per OAuth 2.0 [4] specification.

4. If the CAPIF core function successfully verifies the Access Token Request message, the CAPIF core function shall generate an access token specific to the API invoker and return it in an Access Token Response message.

   Steps 1 to 4 of this procedure may be skipped if the API invoker is already in possession of a valid OAuth access token. In this case, the API invoker begins the procedure at step 5.

NOTE 1: The API invoker may include the CAPIF core function assigned API invoker ID and the Onboard_Secret in the OAuth access token request message for the CAPIF core function to validate the access token request.

NOTE 2: Void.

5. On CAPIF-2e, the API invoker authenticates to the AEF by establishing a TLS session with the API exposing function based on the authentication and authorization method (i.e. Server (AEF) side certificate authentication or certificate-based mutual authentication) as indicated by CAPIF core function. The following procedure shall be performed prior to establishment of TLS session.

   The API invoker shall send Authentication Initiation Request to the AEF, including API invoker ID.

   The AEF shall request for security information from the CAPIF Core Function to perform authentication and secure interface establishment with the API invoker. The CAPIF Core Function provides the security information related to the chosen security method (TLS with OAuth token) to the AEF over CAPIF-3 reference point. The CAPIF core function may return API invoker's root CA certificate for the AEF to validate the API invoker's certificate.

   After fetching the relevant security information for the authentication, the AEF shall send Authentication Initiation Response message to API invoker to initiate the TLS session establishment procedure.

6. With successful authentication to the AEF on CAPIF-2e, the API invoker shall initiate invocation of a 3GPP northbound API with the AEF. The access token received from the CAPIF core shall be sent along with the northbound API invocation request as per OAuth 2.0 [4].

7. The API exposing function shall validate the access token. The AEF verifies the integrity of the access token by verifying the CAPIF core function signature If validation of the access token is successful, the AEF shall verify the API invoker's Northbound API invocation request against the authorization claims in access token, ensuring that the API Invoker has access permission for the requested service API.

8. After successful verification of the access token and authorization claims of the API invoker, the requested northbound API shall be invoked and the appropriate response shall be returned to the API invoker.

6.5.3 Authentication and authorization for RNAA

6.5.3.1 General

The authorization function shall obtain the necessary permission from the resource owner for allowing the API invoker to access a northbound API.

RNAA shall use token-based authorization using OAuth 2.0 framework with the following roles:

- The API invoker has the role of the OAuth 2.0 client.
- The CCF has the role of the OAuth 2.0 authorization server, i.e., providing the access token used for RNAA.
- The AEF has the role of the resource server.

The access tokens used for RNAA shall contain the resource owner ID.

The resource owner may be the user of the UE or the owner of the subscription depending on the use case and regulations.The resource owner ID is specified as the GPSI of the corresponding UE if the resource is related to a UE.

NOTE: The present document does not specify the resource owner.

The access token shall include the resource owner ID and the API invoker ID. The resource owner ID is the GPSI. The API invoker ID binds the token to the API invoker. To avoid privacy issues, GPSI should be different from MSISDN, SUPI etc.

The AEF shall check if the token includes *resOwnerId* claim, which includes resource owner ID, to identify that it is a token used in RNAA.

AEF shall do the authorization check of the API invocation request for accessing the resources of the resource owner. AEF checks the request against the token, including:

1) checking the token integrity and

2) checking whether the GPSI (if present) in the API invocation request is compliant with the resource owner ID in the access token. As the token includes resource owner ID, there is no need for additional UE authentication in API invocation. Moreover, the token should be able to restrict the API invoker to a specific resource (e.g., location, QoS, PDN connectivity status) of the resource owner.

For OAuth 2.0 flows involving redirection, authentication between CCF/AUF and UE should be performed after API Invoker redirects the UE to CCF/AUF.

In case of an external AF (i.e., not the application on the UE) being the API invoker, for mutual authentication of API invoker AF and API exposing function, the authentication methods of clause 6.4 and clause 6.5.2 are reused.

For authorization, the following OAuth 2.0 flows may be used:

- Client credential flow (according to RFC 6749 [4]),
- Authorization code flow (according to RFC 6749 [4]), or
- Authorization code flow with PKCE (according to RFC 7636 [11]).

CCF shall indicate the selected flows to the API invoker.

CCF shall give service authorization which subscribers or users can use RNAA.

For selecting the authorization method, the procedure as specified in clause 6.3.1.2 is used with the following RNAA specific additions. The API invoker shall include in the Security Method Request the supported RNAA authorization flows. The CCF shall determine the RNAA authorization flow based on the RNAA capabilities of the CCF, AEF, and API invoker. The API invoker shall use the determined RNAA authorization flow in the subsequent communication with the CCF and AEF.

NOTE: In the present document, only a UE accessing its own resources is considered if the API invoker is on a UE.

6.5.3.2 Authorization using oauth client credential flow

If client credential flow is used for authorization of the API invoker by the AEF, the procedures in RFC 6749 [4] shall be followed with the following profile:

- The access token request message may include the resource owner ID.

NOTE 1: If the API invoker is on a UE, the CCF obtains its GPSI during authentication.

Editor’s note: the mapping of API Invoker ID and GPSI is left for stage 3.

- The CCF shall check whether the API invoker is entitled to consume the API and allowed to access the resources of the resource owner, by using authorization information available in the CCF.
- If the API invoker is on a UE, the CCF shall check that the UE is accessing its own resources. If the API invoker is an AF not on a UE, the check is omitted.

NOTE 2: How to get the authorization from the resource owner and store it in the CCF is out of scope of the present document.

6.5.3.3 Authorization using authorization code (optional PKCE) flow

If authorization code flow, optionally with PKCE, is used by the AEF for authorization of the API invoker, the procedures in RFC 6749 [4] and optionally RFC 7636 [11] shall be followed, with the following profile:

- The authorization token and/or authorization request may include the resource owner ID.

NOTE: If the API invoker is on a UE, the CCF obtains its GPSI during authentication.

Editor's note: the mapping of API Invoker ID and GPSI is left for stage 3.

- The resource owner dynamically authorizes the API invoker to access the resource owner's resources as described in RFC 6749 [4] and optionally RFC 7636 [11].

- If the API invoker is on a UE, the CCF shall check that the UE is accessing its own resources. The access token shall contain the resource owner ID (i.e. GPSI) and the API invoker ID. If the API invoker is an AF not on a UE, the check is omitted.

6.5.3.4 Revocation

The CCF can initiate the Authorization Revocation Request message as defined in clause 8.23.4 of TS 23.222 [3] with additional information to identify the RNAA-related revoked token.

NOTE: The CCF can receive a revocation request message from the resource owner via the UE, resource owner function, web page etc. All these mechanisms are out of the scope of the present document.

AEF, storing the information about the RNAA-related revoked token, shall check whether the token presented by an API invoker is revoked or not, before responding to the API invoker's invocation request.

The CCF provided notification message to the API invoker shall include the information to identify the RNAA-related revoked token.

6.6 Security procedures for CAPIF-3/4/5 reference points

To ensure security of the interfaces between CAPIF entities within a trusted domain, namely CAPIF-3, CAPIF-4, CAPIF-5:

- TLS shall be used to provide integrity protection, replay protection and confidentiality protection. The support of TLS is mandatory. Security profiles for TLS implementation and usage shall follow the provisions given in TS 33.310 [2], Annex E.

- Certificate based mutual authentication shall be performed between the CAPIF entities using TLS. Certificate based authentication shall follow the profiles given in 3GPP TS 33.310 [2], subclauses 6.1.3a and 6.1.4a. The structure of the PKI used for the certificate is out of scope of the present document.

NOTE: It is up to the domain administrator's policy to protect interfaces within the trusted domain.

6.7 Security procedures for updating security method

As specified in TS 23.222 [3], the CAPIF core function shall receive updates to AEF authentication and authorization method from API publishing function. In case that the AEF updates its authentication and authorization method and API invoker uses the old authentication and authorization method to invoke the service API, the AEF shall send a failure response to the API invoker with an indicator that indicates the authentication and authorization method used by the API invoker is incorrect. The API invoker shall contact the CAPIF core function to get the updated authentication and authorization method. Then the API invoker shall invoke the service API using the updated authentication and authorization method.

6.8 Security procedure for API invoker offboarding

Pre-conditions:

1. The API invoker has been onboarded successfully.

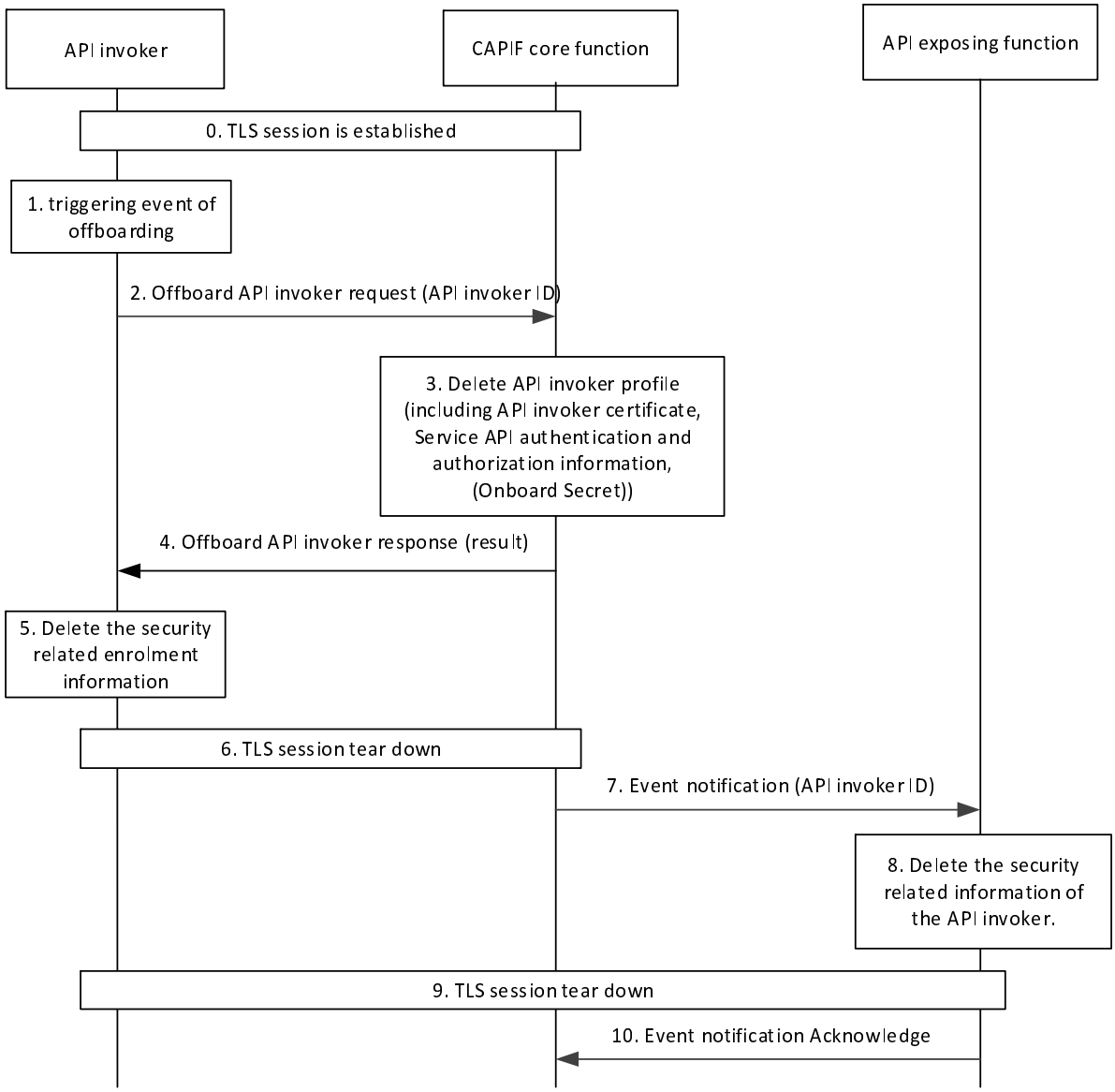

Figure 6.8-1: Security procedure for API invoker offboarding

0. TLS session is established successfully between the CAPIF core function and the API invoker.

1. An event occurs within the API invoker to trigger the offboarding action.

NOTE: The definition of events that trigger offboarding is outside the scope of the present document.

2. The API invoker shall send Offboard API invoker request message to the CAPIF core function, including the CAPIF core function specific API invoker ID which was assigned by the CAPIF core function during the onboarding procedure.

3. The CAPIF core function shall verify the API invoker ID received in step 2 and check that the corresponding profile exists for this API invoker. With successful verification of the API invoker ID and its profile, the CAPIF core function shall cancel the enrolment of the API invoker and delete the API invoker profile. This includes deletion of API invoker certificate, service API authentication and authorization information, and onboard secret (if applicable). Depending on the operator policy, the CAPIF core function may retain the information of the offboarded API invoker.

4. The CAPIF core function sends Offboard API invoker response message, indicating the successful offboarding of the API invoker.

5. The API invoker shall delete the information, such as API invoker ID, Service API authentication / authorization information, API invoker certificate, Onboard_Secret (if applicable).

6. The CAPIF core function shall tear down the TLS session with the API invoker.

7. The CAPIF core function shall send Event notification message to the API exposing function to indicate that this API invoker is no longer valid.

8. The API exposing function shall delete the security related information associated with this API invoker depending on the method that was used previously to authenticate the API invoker, e.g. AEF_{PSK} (TLS-PSK method as described in subclause 6.5.2.1), root certificate to validate the API invoker certificate (PKI method as described in subclause 6.5.2.2), access token (OAuth 2.0 method as described in subclause 6.5.2.3 of the present document, respectively).

9. The API exposing function shall tear down the TLS connection with the API invoker.

10. The API exposing function shall return Event notification acknowledge message to indicate that the security related information associated with this API invoker is successfully deleted and thus the API invoker no longer an acknowledged user.

6.9 Security procedures for CAPIF-7/7e reference points

To ensure security of the interfaces between API Exposing functions (Topology hiding entities and destination AEF handling service APIs), namely CAPIF-7 and CAPIF-7e:

- Security procedures as specified in clause 6.4 of this specification for CAPIF-2 reference point shall be used for secure communication, authentication and authorization, between the AEFs belonging to same trust domain over CAPIF-7 reference point.

- Security procedures as specified in the clause 6.5 of this specification for CAPIF-2e reference point shall be used for secure communication, authentication and authorization, between the AEFs belonging to different trust domains over CAPIF-7e reference point.

6.10 Security procedures for CAPIF-3e/4e/5e reference points

To ensure security of the interfaces between CAPIF entities between different trusted domains (CCF domain and API Provider Domain), namely CAPIF-3e, CAPIF-4e, and CAPIF-5e:

- 3GPP TS 33.210 [10] shall be applied to secure messages on the reference points specified otherwise; and

- 3GPP TS 33.310 [2] may be applied regarding the use of certificates with the security mechanisms of 3GPP TS 33.210 [X] unless otherwise specified in the present document.

SEG as specified in 3GPP TS 33.210 [10] may be used in the trusted domain to terminate the IPsec tunnel.

Annex A (normative): Key derivation functions

A.1 AEF_{PSK} derivation function

AEF_{PSK} key derivation shall be performed using the key derivation function (KDF) specified in TS 33.220 [8]. This subclause specifies how to construct the input string, S, to the KDF (which is input together with the relevant key).

The FC number space is controlled by TS 33.220 [8].

AEF_{PSK} shall be derived by the API invoker and the CAPIF core function based on Service API interface information and CAPIF-1e TLS session parameters. Length and format of TLS session parameters used for key derivation are as specified in TLS. Security profiles for TLS implementation and usage shall follow the provisions given in TS 33.310 [2], Annex E.

The following parameters shall be used to form the input S to the KDF.

FC = 0x7A

P0 = Service API interface information

L0 = Length of Service API interface information

P1 = CAPIF-1e TLS session's Session ID, generated as part of TLS full Handshake.

L1 = Length of TLS Session ID

The input key shall be equal to CAPIF-1e TLS session's Master Secret.

NOTE: Service API interface information is as specified in TS 23.222 [3].

Annex B (informative): Security flows

B.1 Onboarding

Figure B.1-1 shows the functional security flow for online onboarding. Offline onboarding is out of scope for the present document.

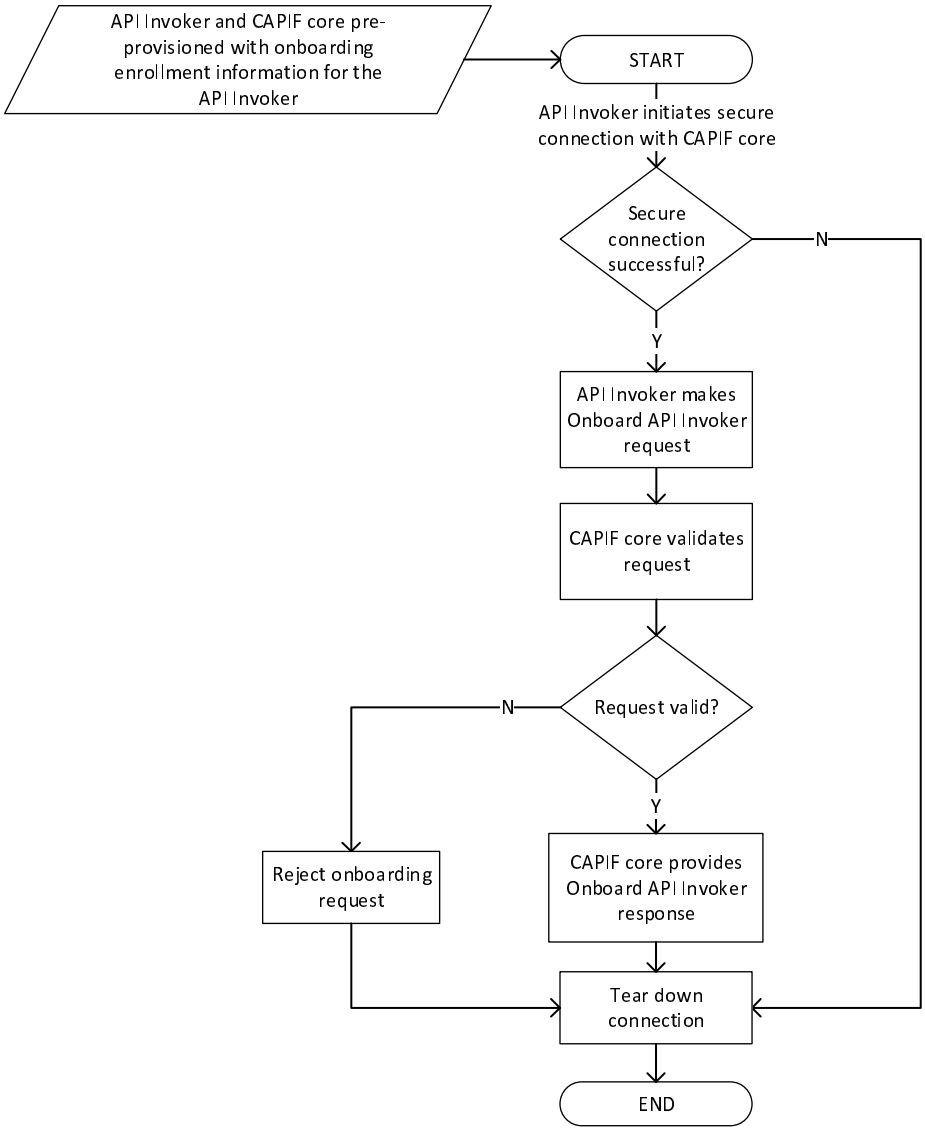

Figure B.1-1: Onboarding security flow

As a pre-requisite to onboarding, the API Invoker and the CAPIF are provisioned with the necessary onboarding enrolment information for the API Invoker. The method to do this is out of scope for the present document.

Initially, the API Invoker attempts to establish a secure connection with the CAPIF core. If the onboarding session cannot be secured, the session is released and the onboarding flow ends.

If the session is secured, the API Invoker requests onboarding using the Onboard API Invoker Request message defined in clause 8.1 of 23.222 [3]. The API Invoker includes an onboarding credential in the Onboard API Invoker Request

message. The CAPIF core receives the Onboard API Invoker request message and validates the onboarding credential. If the onboarding credential is valid, the CAPIF core creates and returns an Onboard API Invoker Response message defined in clause 8.1 of 23.222 [3], which contains the API Invoker profile and includes the API Invoker ID. Security information for CAPIF-1 or CAPIF-1e authentication and (optionally) security information for CAPIF-2 or CAPIF-2e is also transferred to the API Invoker as part of the onboarding response. If the CAPIF core cannot validate the onboarding credentials, then an Onboard API Invoker response message containing an error response is returned to the API Invoker instead.

Following the return of an Onboard API Invoker response message (either successful or unsuccessful), the secure session is torn down and the onboarding security flow ends.

B.2 Authentication and authorization

CAPIF authentication and authorization consists of CAPIF-1e authentication and CAPIF-2e authentication and authorization. Figure B.2-1 shows the functional security flow for CAPIF-1e authentication while Figure B.2-2 shows the functional security flow for CAPIF-2e authentication and authorization.

Prior to starting the security flow for either CAPIF-1e or CAPIF-2e authentication and authorization, successful onboarding of the API Invoker has taken place.

In figure B.2-1, the security flow starts with the API Invoker establishing a TLS connection to the CAPIF core over the CAPIF-1e interface per clause 6.3. Successful TLS establishment results in the opportunity for the CAPIF core to transfer CAPIF-2e AEF authentication and authorization information to the API invoker. After transfer of the CAPIF-2e AEF authentication and authorization information to the API invoker, the TLS session is released and the CAPIF-1e security flow ends.

In the case that either the CAPIF-1e TLS session or API invoker authentication procedure fails, the API Invoker authentication is rejected, AEF authentication and authorization information is not transferred to the API Invoker, and the TLS session with the API Invoker is closed.

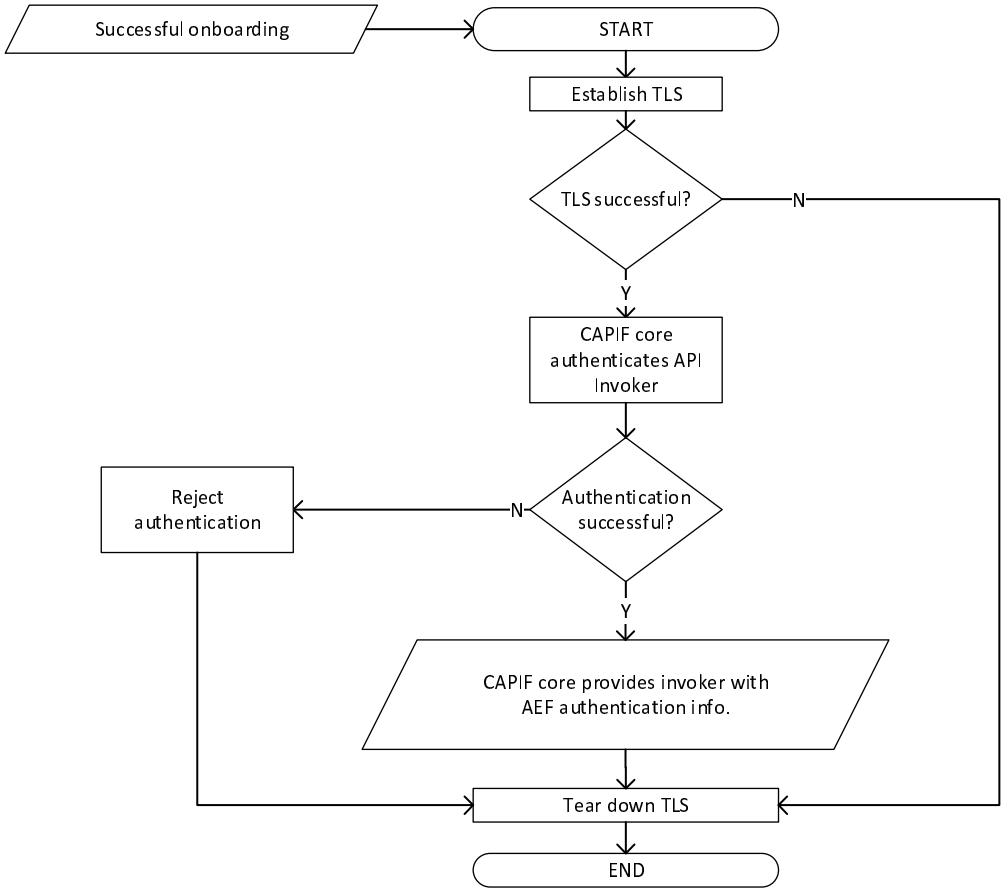

Figure B.2-1: CAPIF-1e authentication

Figure B.2-2 shows the security flow for the CAPIF-2e interface. Successful CAPIF-1e authentication and AEF authentication information (as a minimum) is needed for the API invoker to communicate with the AEF.

The security flow begins when the API Invoker makes an authentication request to the AEF. The AEF receives the request and attempts to authenticate the API Invoker. If the AEF does not possess the authentication information to authenticate the API invoker, the AEF can query the CAPIF core for it. If authentication of the API invoker is successful, then a TLS session is established. If authentication of the API invoker fails, the security flow ends.

If authentication of the API invoker is successful, then based on the interested service API, the API Invoker makes a northbound API request.

The AEF attempts to validate the northbound API request. If the AEF does not possess the authorization information for the requested service API, the AEF can query the CAPIF core for it. If validation of the northbound API request is successful, the northbound API is serviced.

Upon completion of the northbound API action(s), the secure session is torn down and the security flow ends.

If the AEF cannot validate the northbound API request, the AEF rejects the northbound API request, tears down the secure session, and ends the security flow.

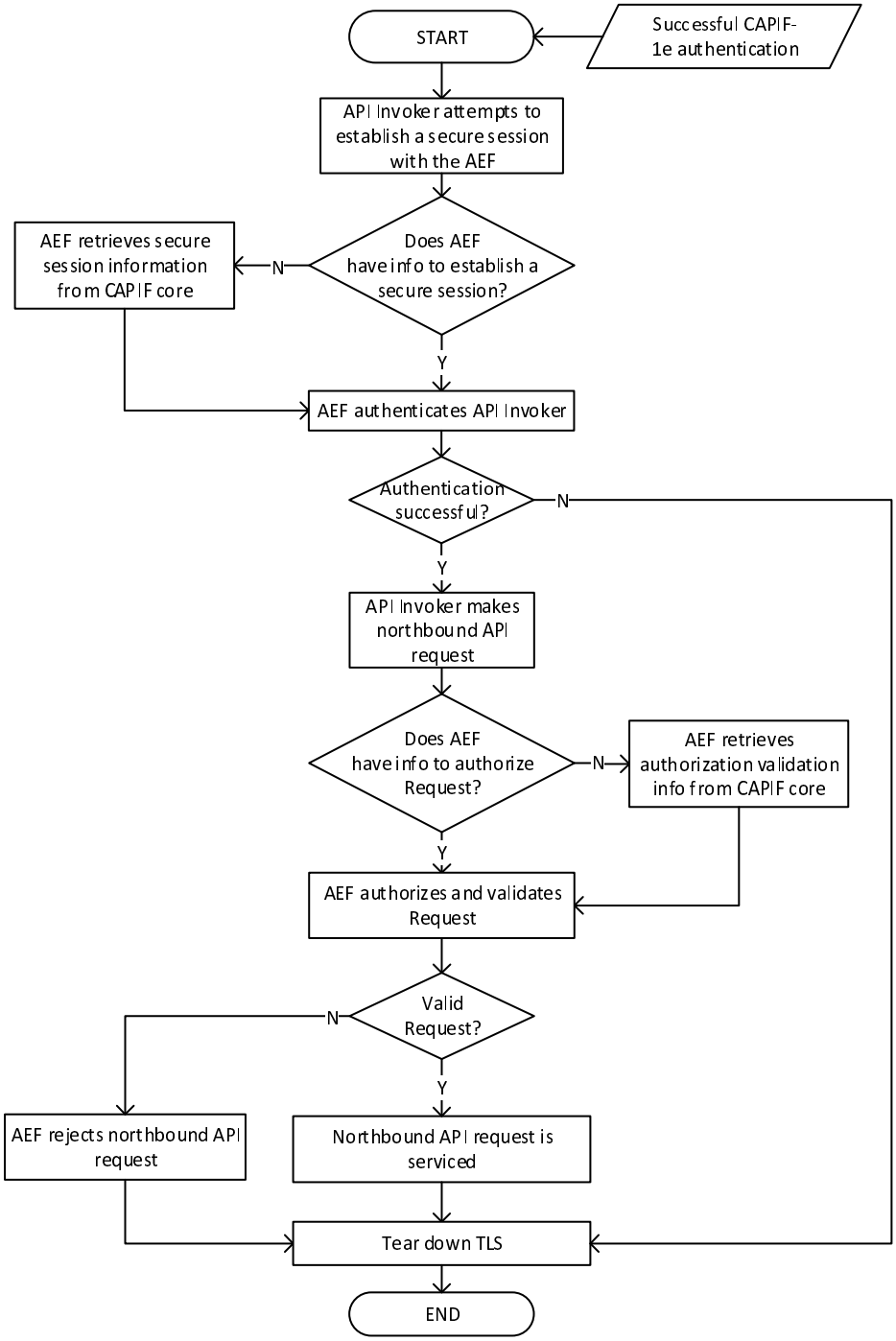

Figure B.2-2: CAPIF-2e authentication and authorization

Annex C (normative): Access token profile

C.1 General

The information in this annex provides a description of two types of access tokens, i.e., the access token used in the 'Method 3 – TLS with OAuth token' authentication and authorisation method (i.e. used for existing CAPIF implementations, see clause 6.5.2.3) and access token used in RNAA (see clause 6.5.3). Characterization of the access token, how to obtain the access token, how to validate the access token, and how to refresh the access token is explained.

An 'Method 3 – TLS with OAuth token' access token or an access token used in RNAA has the following chanracterics:

- Shall be encrypted when transported over the CAPIF 1/1e and CAPIF 2/2e interfaces (e.g. using TLS);
- Shall be a bearer type as specified in IETF RFC 6750 [5];
- Shall be encoded as a JSON Web Token as specified in IETF RFC 7519 [6];
- Shall be protected by the JSON signature profile as specified in IETF RFC 7515 [7]; and,
- Shall be validated per OAuth 2.0 [4], IETF RFC 7519 [6] and IETF RFC 7515 [7].

C.2 Access token profile

C.2.1 General

The 'Method–3 - TLS with OAuth token' access token or an access token used in RNAA contains the token claims described in C.2.2. Token claims of both types of tokens are provided by the CAPIF Core Function and contain authentication and authorization information about the API Invoker. Token claims are used by the API Exposing Function for authorization of API Invoker northbound API requests.

C.2.2 Token claims

The CAPIF 'Method–3 - TLS with OAuth token' access token or an access token used in RNAA shall convey the following claims as defined in IETF RFC 7519 [6] and IETF RFC 6749 [4].

Table C.2.2-1: Access token standard claims

Parameter	Description
exp	REQUIRED. The expiration time of the access token. Implementers MAY provide for some small leeway, usually no more than a few minutes, to account for clock skew (not to exceed 30 seconds).
client_id	REQUIRED. The identifier of the API Invoker making the API request as previously established with the CAPIF Core Function through onboarding.
scope	REQUIRED. A string containing a space-delimited list, comprising of the following as scopes associated with this token: - List of Services per AEF (e.g. "AEF_1:$Service_1$,$Service_2$,$Service_3$,...,$Service_X$; AEF_2:$Service_1$,$Service_2$,$Service_3$,...,$Service_Z$")

The CAPIF OAuth 2.0 access token shall additionally convey the following claim for RNAA.

Table C.2.2-1: Access token customized claims

Parameter	Description
resOwnerId	OPTIONAL. Resource owner ID.

The ‘exp’and ‘scope’ parameters of the access token shall be determined by the CAPIF core function based upon the client_id of the API Invoker provided in the Access Token Request message.

The scope parameter ‘List of Services per AEF’ shall contain a full or partial list of services which the API Invoker is permitted to access at each AEF.

C.3 Obtaining tokens

C.3.1 General

Once an API Invoker has successfully performed onboarding with the CAPIF Core Function, the API Invoker may request the CAPIF ' Method–3 - TLS with OAuth token 'access tokens using ‘Method 3 – TLS with OAuth token’ defined in clause 6.5.2.3 or request access tokens used in RNAA using the methods defined in clause 6.5.3. Figure C.3.1-1 shows the access token request and access token response message exchange.

Figure C.3.1-1: Requesting an access token

NOTE 1: Implementation of the OAuth 2.0 token and authorization endpoints within the CAPIF Core Function are out of scope of this document.

NOTE 2: As described in IETF RFC 6749 [4] clause 4.4, for the CAPIF ' Method–3 - TLS with OAuth token ’ access tokens, the client authentication is used as the authorization grant, therefore no additional authorization request is needed.

C.3.2 Access token request

To obtain an access token, the API Invoker makes a request to the CAPIF Core Function by sending an Access Token Request message with the following parameters using the "application/x-www-form-urlencoded" format, with a character encoding of UTF-8 in the HTTP request entity-body. The access token request parameters are shown in table C.3.2-1.

Table C.3.2-1: Access token request message parameters

Parameter	Values
grant_type	REQUIRED. The value shall be set to "client_credentials or “authorization_code””.
client_id	REQUIRED. The identifier of the API Invoker making the request. It shall match the value that was assigned to the API Invoker during the onboarding process.
client_cred	OPTIONAL. The client credential that was provided to the API Invoker during the onboarding process.
Redirect_uri	OPTIONAL. The value shall be identical with the value in authorization request once authorization code grant or PKCE is used.
code	OPTIONAL. The authorization code received from the CCF for RNAA once authorization code grant or PKCE is used.
code_verifier	OPTIONAL. If the authorization code grant with PKCE flow is selected, the code verifier is used by the CCF to check the code_challenge according to IETF RFC 7636 [11] once PKCE is used.
scope	OPTIONAL. A string containing a space-delimited list, comprising of the following as scopes associated with this token: - List of Services per AEF (e.g. “AEF_1:$Service_1$,$Service_2$,$Service_3$,...,$Service_X$; AEF_2:$Service_1$,$Service_2$,$Service_3$,...,$Service_Z$”)

If the token is used for RNAA (see clause 6.5.3), the parameter resOwnerID is used for the resource owner ID.

resOwnerID	OPTIONAL. Resource owner ID

C.3.3 Access token response

If the access token request (i.e. the client credential) is valid and authorized by the CAPIF Core Function, the CAPIF Core Function then returns an access token to the API Invoker in an access token response message; otherwise it will return an error.

The access token response parameters are shown in table C.3.3-1.

Table C.3.3-1: Access token response message parameters

Parameter	Values
access_token	REQUIRED. This is the issued access token.
Refresh_token	OPTIONAL. This is the issued refresh token.
token_type	REQUIRED. This field shall be “bearer”
expires_in	REQUIRED. The lifetime in seconds of the access token.
scope	OPTIONAL. The granted scope by the CAPIF core function.

Upon receiving the access token reponse message, the API Invoker may now use the access token to make authorized northbound API requests to API Exposure Functions as described in clause 6.5.2.3 or clause 6.5.3.

C.4 Refreshing an access token

C.4.1 Client Credentials Grant

To protect against leakage or other compromise, an access token includes an expiration time. If the API Invoker determines that its access token has expired or if the API Invoker receives an indication from the AEF that a fresh access token is needed, the API Invoker shall return to the CAPIF Core Function and repeat the procedure defined in C.3.

C.4.2 Authorization code grant and PKCE

If the API Invoker determines that its access token has expired or if the API Invoker receives an indication from the AEF that a fresh access token is needed, the API Invoker may use the refresh_token to get a refresh access token as depicted in clause 6 in RFC 6749 [4].

The API Invoker may determine to repeat the procedure defined in C.3 to get a new refresh token and access token.

C.5 Using the token to access API exposing functions

Access tokens of type "bearer" shall be communicated from the API Invoker to AEF by including the access token in the HTTP Authorization Header, per IETF RFC 6750 [5].

The access token is opaque to the API Invoker, meaning that the API Invoker does not have any specific knowledge of the access token itself. The API Invoker shall use the ‘expires_in’ parameter from the access token response message to determine whether the access token is valid so that it does not send an expired access token to AEFs. If the access token is presented to an AEF and the token is expired or revoked, the AEF should return an error message indicating such to the API Invoker.

C.6 Token revocation

In order to limit the time validity of a token, the "exp" and "expires_in" parameters shall be used as a method of access token revocation.

Within the claims of a ‘Method 3 - TLS with OAuth token’ access token, the "exp" parameter shall be used by the AEF to determine whether or not the token has expired. If the current time is beyond the time specified by the "exp" parameter, the associated token shall no longer be considered valid and any requests made with an expired token shall be rejected by the AEF.

Within the claims of an access token response message, the "expires_in" parameter shall be used by the API Invoker to determine validity of the associated token. If the current time is beyond the time specified by the "expires_in" parameter, the associated access token shall no longer be considered valid and no northbound API requests shall be made using the expired access token. The procedure defined in C.3 shall be used to obtain a new access token.

C.7 Token validation

C.7.1 Access token validation

A non-RNAA access token, i.e. ‘Method 3 – TLS with OAuth token’ access token shall be validated according to IETF RFC 7519 [6].

Annex D (informative): Change history

Change history							
Date	Meeting	TDoc	CR	Rev	Cat	Subject/Comment	New version
2018-06	SA#80	SP-180461				Presented for information and approval	1.0.0
2018-06	SA#80					Upgrade to change control version	15.0.0
2018-09	SA#81	SP-180699	0001	1	B	[CAPIF-Sec] 33122 CAPIF access token definition	15.1.0
2018-09	SA#81	SP-180699	0003	1	F	[CAPIF-Sec] 33122 correct note in clause 6.5.2.1	15.1.0
2018-09	SA#81	SP-180699	0005	1	B	Security requirements for API invoker onboarding and offboarding	15.1.0
2018-09	SA#81	SP-180699	0006	1	F	Modifications on Security procedures for API invoker onboarding	15.1.0
2018-09	SA#81	SP-180699	0007	1	F	Clarification on Security protections in CAPIF-1 and CAPIF-2 reference point	15.1.0
2018-09	SA#81	SP-180699	0008	1	F	Clarification on access token verification	15.1.0
2018-09	SA#81	SP-180699	0009	1	F	Adding FC value to TS 33.122	15.1.0
2018-12	SA#82	SP-181029	0013	1	F	Correction/enhancement in CAPIF TS	15.2.0
2018-12	SA#82	SP-181029	0014	1	F	Delete information during API invoker offboarding	15.2.0
2018-12	SA#82	SP-181029	0017	-	F	Missing subclause headings	15.2.0
2019-03	SA#83	SP-190097	0018	-	F	Editorial corrections in CAPIF TS	15.3.0
2019-06	SA#84	SP-190362	0019	-	B	Security Requirements for CAPIF-3e/4e/5e reference points	16.0.0
2019-09	SA#85	SP-190678	0022	1	B	Security aspects of CAPIF-7/7e reference points	16.1.0
2019-09	SA#85	SP-190678	0023	-	D	Editorial correction of CAPIF-3e/4e/5e requirements clause.	16.1.0
2019-09	SA#85	SP-190678	0024	-	B	Security procedures for CAPIF-7/7e reference points	16.1.0
2019-09	SA#85	SP-190678	0025	-	B	Security procedures for CAPIF-3e/4e/5e reference points	16.1.0
2019-12	SA#86	SP-191133	0026	1	B	Description of CAPIF reference point: 3e,4e,5e,7 and 7e	16.2.0
2020-07	SA#88E	SP-200359	0027	-	B	Usage of TLS profiles for CAPIF	16.3.0
2022-03	-	-	-	-	-	Update to Rel-17 version (MCC)	17.0.0
2022-12	SA#98e	SP-221149	0033	1	A	Correcting the OAuth 2.0 roles in CAPIF	17.1.0
2023-06	SA#100	SP-230599	0034	1	F	CAPIF-2e interface authentication and protection if used by non-3GPP AEFs	18.0.0
2023-09	SA#101	SP-230900	0035	1	F	CAPIF-2e interface authentication and protection if used by non-3GPP AEFs	18.1.0
2023-09	SA#101	SP-230907	0036	-	B	Security for resource owner aware northbound access to APIs	18.1.0
2023-12	SA#102	SP-231340	0039	1	F	Clarification on authentication and authorization for RNAA	18.2.0
2023-12	SA#102	SP-231340	0041	1	F	Access token profile for RNAA	18.2.0
2023-12	SA#102	SP-231340	0046	1	F	Clarification on resource owner ID	18.2.0
2023-12	SA#102	SP-231340	0050	1	F	Detailed functional security model description for support of RNAA	18.2.0
2023-12	SA#102	SP-231340	0051	1	F	Clarification for CAPIF-8	18.2.0
2023-12	SA#102	SP-231340	0052	1	F	Resolve EN related to authorization flow	18.2.0
2024-03	SA#103	SP-240370	0059	-	F	Correction on authentication and authorization for RNAA	18.3.0
2024-03	SA#103	SP-240370	0060	1	F	Access token details	18.3.0
2024-03	SA#103	SP-240370	0061	1	F	Clarification to flow selection for RNAA	18.3.0
2024-03	SA#103	SP-240370	0062	1	F	Alignment of 33.122 for RNAA	18.3.0
2024-03	SA#103	SP-240370	0063	-	F	Update to RNAA functional security model description	18.3.0
2024-03	SA#103	SP-240370	0067	1	F	Update for CAPIF 8	18.3.0
2024-03	SA#103	SP-240370	0068	1	F	Add revocation procedure for RNAA-related tokens	18.3.0
2024-03	SA#103	SP-240370	0070	-	F	Resolve EN related to authorization request or token request	18.3.0
2024-06	SA#104	SP-240655	0072	-	F	Resource owner function	18.4.0
2024-06	SA#104	SP-240655	0074	1	F	Corrections and removing token claim related EN	18.4.0

History

Document history		
V18.3.0	May 2024	Publication
V18.4.0	July 2024	Publication