ETSI TS 123 222 V18.5.0 (2024-07)

LTE;
5G;
Common API Framework for 3GPP Northbound APIs
(3GPP TS 23.222 version 18.5.0 Release 18)

Reference
RTS/TSGS-0623222vi50

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.

Introduction

In 3GPP, there are multiple northbound API-related specifications (e.g. APIs for Service Capability Exposure Function (SCEF) functionalities defined in 3GPP TS 23.682 [2], API for the interface between MBMS service provider and BM-SC defined in 3GPP TR 26.981 [5]). To avoid duplication and inconsistency of approach between different API specifications, 3GPP has considered the development of a common API framework (CAPIF) that includes common aspects applicable to any northbound service APIs.

The present document specifies the functional model, procedures and information flows needed to support the CAPIF, and the guidelines for consistent northbound API (service and CAPIF APIs) development in 3GPP.

NOTE: It is possible to use the CAPIF defined common aspects for other APIs as well, apart from northbound APIs.

1 Scope

The present document specifies the architecture, procedures and information flows necessary for the CAPIF. The aspects of this specification include identifying architecture requirements for the CAPIF (e.g. registration, discovery, identity management) that are applicable to any service APIs when used by northbound entities, as well as any interactions between the CAPIF and the service APIs themselves. The common API framework applies to both EPS and 5GS, can be hosted within a PLMN or SNPN, and is independent of the underlying 3GPP access (e.g. E-UTRA, NR).

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 23.682: "Architecture enhancements to facilitate communications with packet data networks and applications".

[3] 3GPP TS 23.501: "System Architecture for the 5G System; Stage 2".

[4] 3GPP TS 23.502: "Procedures for the 5G System; Stage 2".

[5] 3GPP TR 26.981: "MBMS Extensions for Provisioning and Content Ingestion".

[6] 3GPP TS 32.240: "Telecommunication management; Charging management; Charging architecture and principles".

[7] ETSI GS MEC 011 (V1.1.1): "Mobile Edge Computing (MEC); Mobile Edge Platform Application Enablement".

[8] ETSI GS MEC 009 (V1.1.1): "Mobile Edge Computing (MEC); General Principles for Mobile Edge Service APIs".

[9] OMA-ER_Autho4API-V1_0-20141209-A: "Authorization Framework for Network APIs".

[10] OMA-TS-REST_NetAPI_Common-V1_0-20180116-A: "Common definitions for RESTful Network APIs".

[11] OMA-TS-NGSI_Registration_and_Discovery-V1_0-20120529-A: "NGSI Registration and Discovery".

[12] 3GPP TS 33.122: "Security Aspects of Common API Framework for 3GPP Northbound APIs".

[13] IETF RFC 6749 (October 2012): "The OAuth 2.0 Authorization Framework".

[14] 3GPP TS 23.434: "Service Enabler Architecture Layer for Verticals (SEAL); Functional architecture and information flows".

3 Definitions 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].

API: The means by which an API invoker can access the service.

API invoker: The entity which invokes the CAPIF or service APIs.

API invoker profile: The set of information associated to an API invoker that allows that API invoker to utilize CAPIF APIs and service APIs.

API exposing function: The entity which provides the service communication entry point for the service APIs.

API exposing function location: The location information (e.g. civic address, GPS coordinates, data center ID) where the API exposing function providing the service API is located.

API category: A name used to group service APIs by the service domain in which they are used (e.g., IoT, V2X). The CAPIF core function maintains a list of API category names.

CAPIF administrator: An authorized user with special permissions for CAPIF operations.

Common API framework: A framework comprising common API aspects that are required to support service APIs.

Designated CAPIF core function: The CAPIF core function which is configured as the serving CAPIF core function for interconnection.

Northbound API: A service API exposed to higher-layer API invokers.

Onboarding: One time registration process that enables the API invoker to subsequently access the CAPIF and the service APIs.

Resource: The object or component of the API on which the operations are acted upon.

Resource owner: An entity (either a UE user or an MNO subscriber) capable of granting access to a protected resource related to the invoked API.

Resource owner-aware northbound API access: An API invocation scenario where the API invoker needs an authorization from the resource owner.

Service API: The interface through which a component of the system exposes its services to API invokers by abstracting the services from the underlying mechanisms.

Serving Area Information: The location information for which the service APIs are being offered to.

CAPIF provider domain: A domain that contains an instance of CAPIF core function and may contain API provider domains and API invokers. The CAPIF provider could be a PLMN, SNPN or 3rd party. Throughout this document, PLMN trust domain is often used as the typical deployment of a CAPIF provider domain however SNPN trust domain or 3rd party trust domain are applicable as well.

PLMN trust domain: The entities protected by adequate security and controlled by the PLMN operator or a trusted 3rd party of the PLMN.

SNPN trust domain: The entities protected by adequate security and controlled by the SNPN operator or a trusted 3rd party of the SNPN. **3rd party trust domain:** The entities protected by adequate security and controlled by the 3rd party.

For the purposes of the present document, the following terms and definitions given in 3GPP TS 32.240 [6] apply:

Offline charging
Online charging

3.2 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].

5GS	5G System
AEF	API Exposing Function
AF	Application Function
AMF	API Management Function
APF	API Publishing Function
API	Application Program Interface
AS	Application Server
BM-SC	Broadcast Multicast Service Centre
CAPIF	Common API Framework
CDR	Charging Data Record
CRUD	Create, Read, Update, Delete
DDoS	Distributed Denial of Service
E-UTRA	Evolved Universal Terrestrial Radio Access
EPS	Evolved Packet System
ETSI	European Telecommunications Standards Institute
GS	Group Specification
IP	Internet Protocol
MBMS	Multimedia Broadcast and Multicast Service
MEC	Multi-access Edge Computing
NEF	Network Exposure Function
NGSI	Next Generation Service Interfaces
NR	New Radio
OMA	Open Mobile Alliance
OAM	Operations, Administration and Maintenance
OWSER	OMA Web Services
PC	Protocol Converter
PLMN	Public Land Mobile Network
REST	REpresentational State Transfer
RNAA	Resource owner-aware Northbound API Access
RPC	Remote Procedure Call
SCEF	Service Capability Exposure Function
SCS	Service Capability Server
SNPN	Stand-alone Non-Public Network
UDDI	Universal Description, Discovery and Integration
URI	Uniform Resource Identifier
WSDL	Web Services Description Language

4 Architectural requirements

4.1 General

4.1.1 Introduction

This subclause specifies the general requirements for CAPIF architecture.

4.1.2 Requirements

[AR-4.1.2-a] The CAPIF shall provide mechanisms (e.g. publish service APIs, authorization, logging, charging) to support service API operations.

[AR-4.1.2-b] The CAPIF shall enable API invoker(s) to discover and communicate with service APIs from the API providers.

[AR-4.1.2-c] Reference points between CAPIF and external applications shall be provided as APIs.

[AR-4.1.2-d] Reference points internal to CAPIF may be provided as APIs.

4.1.3 Requirements for supporting 3rd party API providers

[AR-4.1.3-a] The CAPIF shall provide mechanisms (e.g. publish service APIs, authorization, logging, charging) to support service API operations from trusted 3rd party API providers.

[AR-4.1.3-b] The CAPIF shall enable API invoker(s) to discover and communicate with service APIs from trusted 3rd party API providers.

4.2 Service API publish and discover

4.2.1 Introduction

This subclause specifies the service API publish and discover related requirements.

4.2.2 Requirements

[AR-4.2.2-a] The CAPIF shall provide a mechanism to publish the service API information to be used by the API invokers to discover and subsequently invoke the service API.

[AR-4.2.2-b] The CAPIF shall provide a mechanism for the API invokers to discover the published service API information as specified in [AR-4.2.2-a] according to the API invokers' interest.

[AR-4.2.2-c] The CAPIF shall provide a mechanism to restrict the discovery of the published service API information by the API invokers, based on configured policies.

[AR-4.2.2-d] The CAPIF shall provide a mechanism to configure policies to restrict the discovery of the published service API information.

[AR-4.2.2-e] The CAPIF shall provide mechanism to support Serving Area Information related to service APIs.

4.2.3 Requirements for 3rd party API providers

[AR-4.2.3-a] The CAPIF shall provide a mechanism to publish the service API information of the 3rd party API providers.

4.3 Security

4.3.1 Introduction

This subclause specifies the security related requirements for API invokers.

4.3.2 Requirements

[AR-4.3.2-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.

[AR-4.3.2-b] The CAPIF shall provide mechanisms to authenticate API invokers prior to accessing the service APIs.

[AR-4.3.2-c] The CAPIF shall provide mechanisms to authenticate API invokers upon the service API invocation.

[AR-4.3.2-d] The CAPIF shall provide mechanisms to authorize API invokers to access the service APIs.

[AR-4.3.2-e] The CAPIF shall provide mechanisms to validate authorization of the API invokers upon the service API invocation.

[AR-4.3.2-f] The CAPIF shall provide mechanisms for mutual authentication between the CAPIF and the API invoker.

[AR-4.3.2-g] The CAPIF shall provide mechanisms to control the service API access for every API invocation.

[AR-4.3.2-h] The communication between the CAPIF and the API invoker shall be confidentiality protected.

[AR-4.3.2-i] The communication between the CAPIF and the API invoker shall be integrity protected.

[AR-4.3.2-j] The CAPIF shall provide mechanisms to authenticate the service API publishers to publish and manage the service API information.

[AR-4.3.2-k] The CAPIF shall provide mechanisms to authorize the service API publishers to publish and manage service API information.

[AR-4.3.2-l] The CAPIF shall provide mechanisms to validate authorization of the service API publishers to publish and manage service API information.

4.3.3 Additional requirements for 3rd party API provider

[AR-4.3.3-a] 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.

[AR-4.3.3-b] The CAPIF shall provide authorization mechanism for service APIs from the 3rd party API providers.

[AR-4.3.3-c] The CAPIF shall provide data confidentiality (across API providers) for data (e.g. logging, charging) related to service APIs from multiple API providers.

4.4 Charging

4.4.1 Introduction

This subclause specifies the charging related requirements for the usage of service APIs.

4.4.2 Requirements

[AR-4.4.2-a] The CAPIF shall support online and offline charging for service APIs usage.

[AR-4.4.2-b] The CAPIF shall provide mechanisms to record the usage (e.g. invocation count) of the service APIs for charging purpose, on a per API invoker basis.

[AR-4.4.2-c] The CAPIF shall provide mechanisms to record timestamp of the service API invocation.

[AR-4.4.2-d] The CAPIF shall provide mechanisms to record the service API related information, e.g. API location.

4.4.3 Requirements for 3rd party API providers

[AR-4.4.3-a] The CAPIF shall support online and offline charging for 3rd party API providers' service APIs usage.

[AR-4.4.3-b] The CAPIF shall provide mechanisms to query charging related information of the 3rd party service APIs by the authorized users.

4.5 Operations, Administration and Maintenance

4.5.1 Introduction

This subclause specifies the OAM aspects including performance monitoring, fault monitoring, policy configurations, and certain lifecycle management aspects such as monitoring the running status of service APIs and related operations.

4.5.2 Requirements

[AR-4.5.2-a] The CAPIF shall provide mechanisms to monitor the status of service APIs, e.g. starting and stopping access of the service APIs.

[AR-4.5.2-b] The CAPIF shall provide mechanisms to monitor and report the performance of the service APIs.

[AR-4.5.2-c] The CAPIF shall provide mechanisms to monitor and report the fault information about the service APIs.

[AR-4.5.2-d] The CAPIF shall provide mechanisms to record change events of service APIs, e.g. service APIs relocation.

[AR-4.5.2-e] The CAPIF shall provide mechanisms to configure policies related to service APIs.

4.5.3 Requirements for 3rd party API providers

[AR-4.5.3-a] The CAPIF shall provide mechanisms to configure policies related to 3rd party service APIs by the authorized users.

[AR-4.5.3-b] The CAPIF shall provide mechanisms to monitor faults, performance and status of the 3rd party service APIs by the authorized users.

4.6 Service API invocation monitoring

4.6.1 Introduction

The CAPIF includes monitoring functions. This enables API provider to monitor service API invocations, to determine critical aspects such as system load, API usage information, uncover potential overload and attacks (e.g. DDoS) conditions.

4.6.2 Requirements

[AR-4.6.2-a] The CAPIF shall provide mechanisms to capture service API invocation events and make them available to service API provider.

[AR-4.6.2-b] The CAPIF shall provide mechanisms to notify events related to overload and threat conditions (e.g. system load, resource usage information).

[AR-4.6.2-c] The CAPIF shall provide mechanisms to allow service API provider to apply monitoring filters based on criteria such as API invoker's ID and IP address, service API name and version, invoked operation, input parameters, and invocation result.

4.7 Logging

4.7.1 Introduction

The CAPIF supports the ability to log events and store the corresponding logs. This enables the API providers to use the logs for the purpose of tracing back and statistical analysis.

The following events in CAPIF are supported for logging:

- Service API invocation events;
- API invoker onboarding events; and
- API invoker interactions with the CAPIF (e.g. authentication, authorization, discover service APIs).

4.7.2 Logging events related to service API invocations

[AR-4.7.2-a] The CAPIF shall provide mechanisms for service API invocation event logging and storage functionality.

[AR-4.7.2-b] The service API invocation log shall be stored for a configurable time period, according to the service API provider's policy.

[AR-4.7.2-c] The service API invocation log shall be stored securely, and shall only be accessed by CAPIF administrators of the service API provider.

4.7.3 Logging events related to API invoker onboarding

[AR-4.7.3-a] The CAPIF shall provide mechanisms for API invoker onboarding event logging and storage functionality.

[AR-4.7.3-b] The API invoker onboarding log shall be stored at least for the duration during which the onboarding is valid.

[AR-4.7.3-c] The API invoker onboarding log shall be stored securely, and shall only be accessed by CAPIF administrators.

4.7.4 Logging events related to API invoker interaction with the CAPIF

[AR-4.7.4-a] The CAPIF shall provide mechanisms for the event logging of API invoker interactions with the CAPIF (e.g. authentication, authorization, discover service APIs).

[AR-4.7.4-b] The API invoker interactions log shall be stored for a configurable time period.

[AR-4.7.4-c] The API invoker interactions log shall be stored securely, accessed only by CAPIF administrators.

4.8 Auditing service API invocation

4.8.1 Introduction

The CAPIF includes auditing capabilities. This enables the service API provider to identify illegal service API invocations e.g. by querying the service API invocation log.

4.8.2 Requirements

[AR-4.8.2-a] The CAPIF shall provide mechanisms to query the service API invocation log, by CAPIF administrators.

4.9 Onboarding API invoker

4.9.1 Introduction

This subclause specifies the requirements related to onboarding API invoker to the CAPIF.

4.9.2 Requirements

[AR-4.9.2-a] The CAPIF shall provide the capability to onboard new API invokers.

[AR-4.9.2-b] The CAPIF shall support granting an API invoker's request to onboard with the CAPIF administrator.

[AR-4.9.2-c] The CAPIF shall support offboarding an API invoker from the CAPIF.

[AR-4.9.2-d] The CAPIF shall support updating an API invoker's API list e.g., subsequent to discovery of new API(s).

4.10 Policy configuration

4.10.1 Introduction

This subclause specifies the policy configuration related requirements.

4.10.2 Requirements

[AR-4.10.2-a] The CAPIF shall support policy configurations (e.g. related to the protection of platforms and network, specific functionalities exposed, message payload size or throughput).

4.11 Protocol design

4.11.1 Introduction

In order for the CAPIF to be common across all present and future API invokers for various usages and purposes, a minimum common protocol stack model is necessary so that all API invokers that use the common-framework-based API need to support only one and the same set of protocols, e.g. security layer protocol(s). Extensibility of this model allows evolution and re-use.

4.11.2 Requirements

[AR-4.11.2-a] The CAPIF shall support a minimum common protocol stack model common for all API implementations to be based on.

[AR-4.11.2-b] The CAPIF shall support a common security mechanism for all API implementations to provide confidentiality and integrity protection.

[AR-4.11.2-c] The CAPIF shall be extensible to support different protocol stack models, including related security mechanisms, in addition to the minimum common protocol stack model.

NOTE: Potentially, Stage 3 needs to consider all CAPIF APIs for protocol extensibility.

[AR-4.11.2-d] CAPIF APIs and associated information flows shall be extensible to support vendor-specific functionality.

4.12 Interconnection between the CAPIF providers

4.12.1 Introduction

Two organizations with a business relationship that have each deployed CAPIF may need to interoperate to allow API invokers in each trust domain to utilize service APIs from both CAPIFs as illustrated in figure 4.12.1-1.

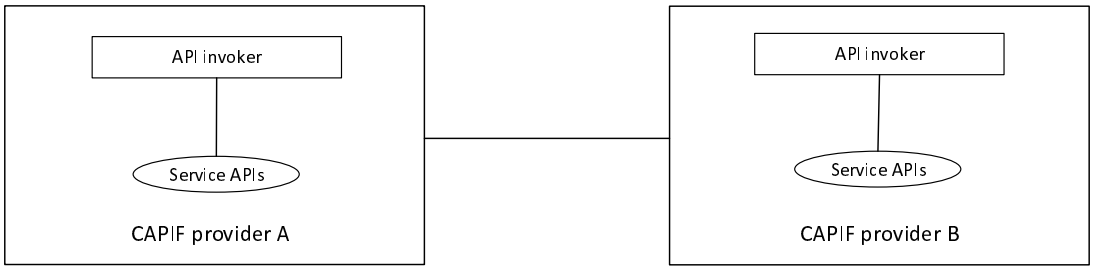

Figure 4.12.1-1: Interconnection between the CAPIF providers

NOTE: From each CAPIF provider's perspective the other CAPIF provider is a 3rd party.

4.12.2 Requirements

[AR-4.12.2-a] The CAPIF shall provide mechanisms to enable the API invokers of the CAPIF provider to discover and invoke the service APIs of the 3rd party CAPIF provider.

4.13 Identities

4.13.1 Introduction

This subclause specifies the identities related requirements.

4.13.2 Requirements

[AR-4.13.2-a] The CAPIF shall support uniform addressing (e.g. identity) for communication within the same trust domain or from the 3rd party trust domain.

[AR-4.13.2-b] The CAPIF shall support identities for uniquely identifying each API.

4.14 API provider domain interactions

4.14.1 Introduction

This subclause specifies the API provider domain interactions related requirements.

4.14.2 Requirements

[AR-4.14.2-a] The CAPIF shall enable interactions between multiple API exposing functional entities within the same trust domain.

[AR-4.14.2-b] The CAPIF shall enable interactions of multiple API exposing functional entities between trust domains.

Editor's note: Adding architectural requirements for interactions between other functions within the API provider domain is FFS.

4.15 Dynamic routing of service API invocation

4.15.1 Introduction

This subclause specifies the dynamic routing of service API invocation related requirements.

4.15.2 Requirements

[AR-4.15.2-a] The CAPIF shall provide a mechanism to support the dynamic routing of service API invocation.

4.16 Registering API provider domain functions

4.16.1 Introduction

This subclause specifies the requirements related to registration of API provider domain functions on the CAPIF core function.

4.16.2 Requirements

[AR-4.16.2-a] The CAPIF shall provide the capability to register API provider domain functions.

[AR-4.16.2-b] The CAPIF shall support the capability to update the registration information of the API provider domain functions.

4.17 Resource owner-aware northbound API invocation

4.17.1 Introduction

This subclause specifies requirements related to the resource owner-aware northbound API invocation. In the current release, the scope of API invoker on a UE in Resource owner-aware northbound API access is limited to accessing its own resources only, i.e., resource owner is a user of the UE hosting the API invoker that can authorize the API access.

4.17.2 Requirements

[AR-4.17.2-a] The CAPIF shall support applications on the UE acting as an API invoker.

[AR-4.17.2-b] The CAPIF shall support the authentication of the resource owner.

[AR-4.17.2-c] The CAPIF shall enable the resource owner(s) to provide and revoke the authorization information for the resource exposure by API provider.

5 Involved business relationships

5.1 Basic CAPIF business relationships

Figure 5.1-1 shows the typical business relationships in CAPIF.

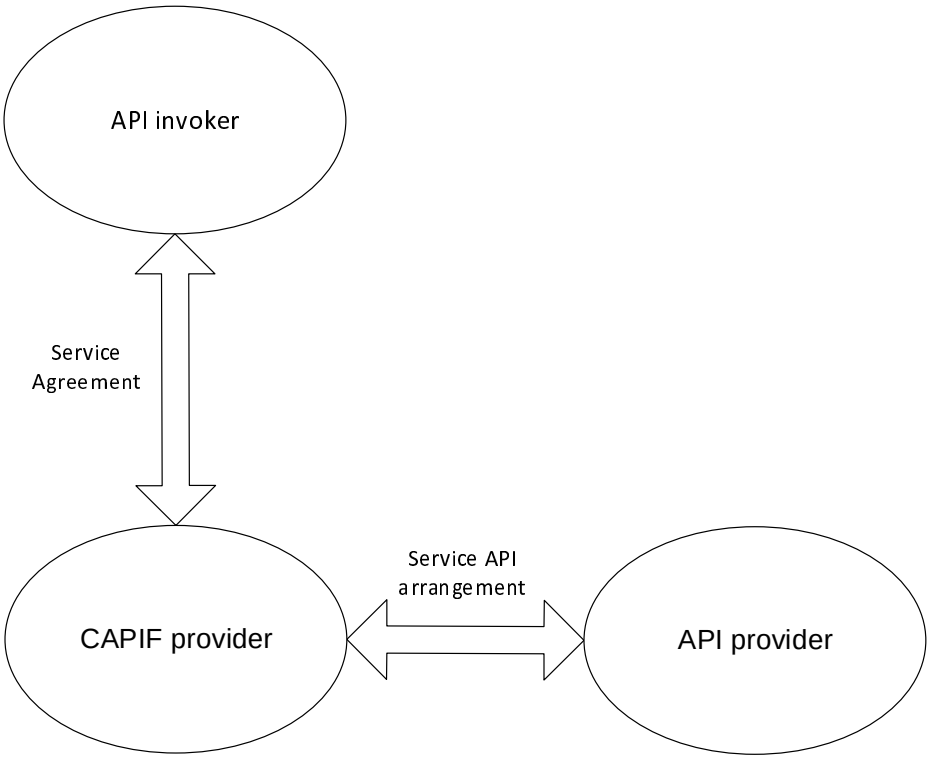

Figure 5.1-1: Business relationships in CAPIF

The API invoker is typically provided by a 3rd party application provider who has service agreement with a CAPIF provider.

The API provider hosts one or more service APIs and has a service API arrangement with CAPIF provider to offer the service APIs to the API invoker.

The CAPIF provider and the API provider can be part of the same organization (e.g. PLMN operator), in which case the business relationship between the two is internal to a single organization. The CAPIF provider and the API provider can be part of different organizations, in which case the business relationship between the two must exist.

5.2 CAPIF business relationships for RNAA

Figure 5.2-1 shows the CAPIF business relationships for the resource owner-aware northbound API access (RNAA).

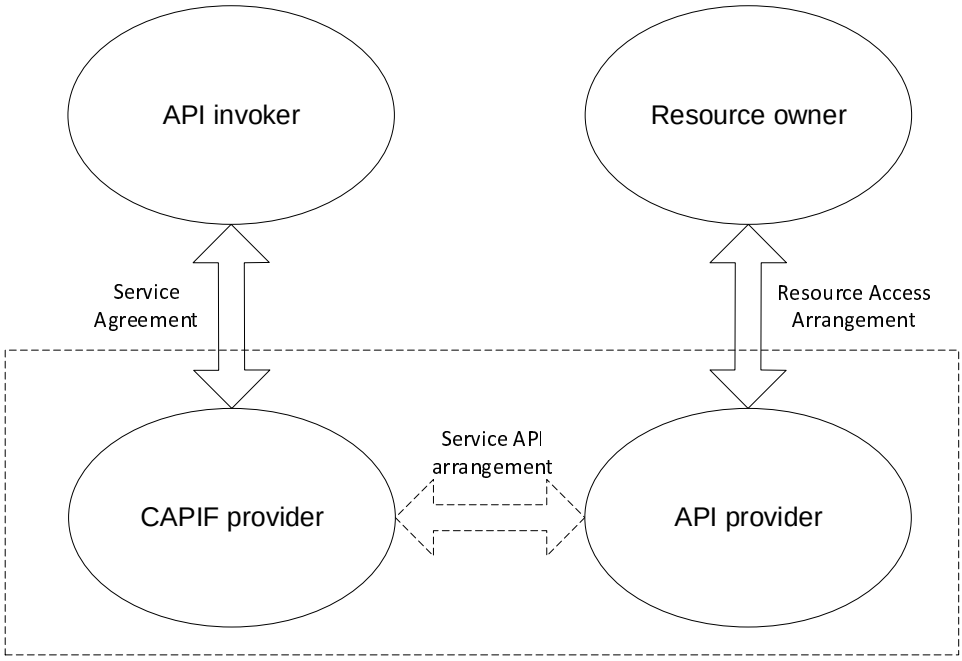

Figure 5.2-1: CAPIF business relationships for RNAA

The business relationships the API invoker, the CAPIF provider, and the API provider follow the description in the clause 5.1. In addition to them, the resource owner is an entity capable of granting access to a protected resource related to the resource exposed by the API provider. The API invoker and the resource owner can be the same entity or separate entities. In the current release, the resource owner is a user of a UE and can provide authorization information using the UE.

NOTE: In the current release, both the CAPIF provider and the API provider should belong to the same organization (e.g., PLMN operator) and the service API arrangement is not required explicitly.

6 Functional model

6.1 General

The Common API framework (CAPIF) functional architecture is described in this subclause. The CAPIF architecture is defined as service-based and interactions between the CAPIF functions are represented in two ways:

- A service-based representation, where CAPIF functions enable other authorized CAPIF functions to access their services;

- A reference point representation, where interactions between any two CAPIF functions (e.g. CCF, AEF) is shown by an appropriate point-to-point reference point (e.g. CAPIF-3).

The CAPIF functional architecture can be adopted by any 3GPP functionality providing 3GPP northbound service APIs.

NOTE 1: The terms “functional architecture” and “functional model” mean the same and have been used interchangeably in this specification.

NOTE 2: The functional model described in this specification applies to both PLMN(s) and to SNPN(s).

6.2 Functional model description

6.2.0 Functional model description for the CAPIF

Figure 6.2.0-1 shows the reference point based functional model for the CAPIF.

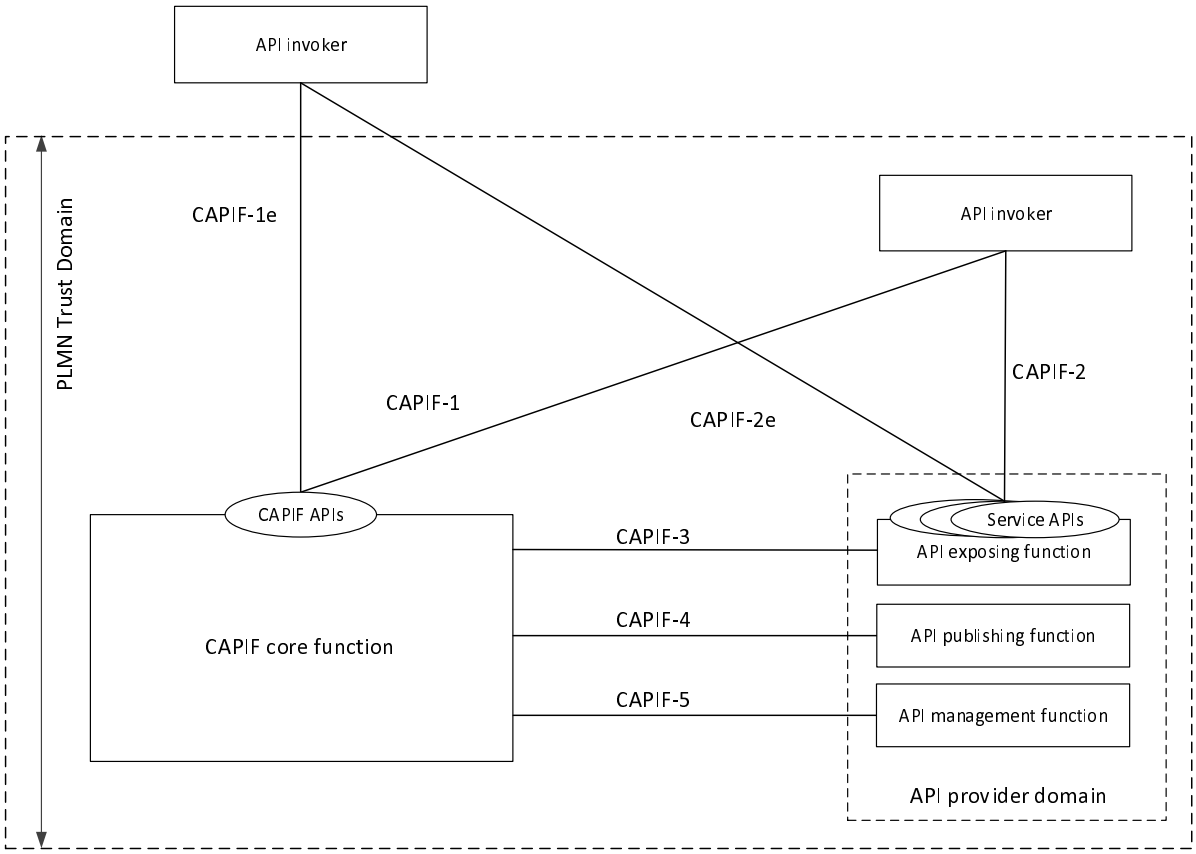

Figure 6.2.0-1: Functional model for the CAPIF

The CAPIF is hosted within the PLMN operator network (or even an SNPN). The API invoker is typically provided by a 3rd party application provider who has service agreement with PLMN operator. The API invoker may reside within the same trust domain as the PLMN operator network.

In a reference point based model, the API invoker within the PLMN trust domain interacts with the CAPIF via CAPIF-1 and CAPIF-2. The API invoker from outside the PLMN trust domain interacts with the CAPIF via CAPIF-1e and CAPIF-2e. The API exposing function, the API publishing function and the API management function of the API provider domain (together known as API provider domain functions) within the PLMN trust domain interacts with the CAPIF core function via CAPIF-3, CAPIF-4 and CAPIF-5 respectively.

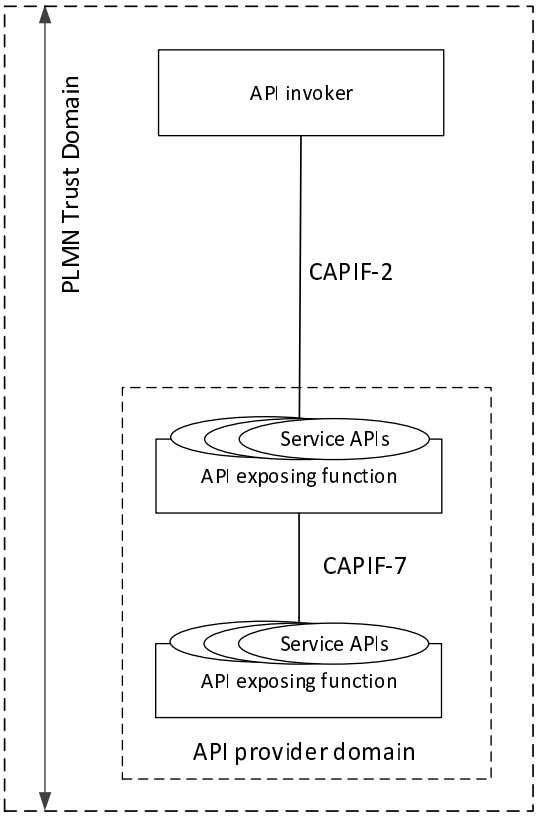

Figure 6.2.0-2: Functional model for interactions between API exposing functions

As illustrated in figure 6.2.0-2, the interactions between the API exposing functions within the PLMN trust domain is via CAPIF-7.

The CAPIF core function provides CAPIF APIs to the API invoker over CAPIF-1 and CAPIF-1e. The API exposing function provides the service APIs to the API invoker over CAPIF-2 and CAPIF-2e.

NOTE 1: The communication between the API exposing function and the CAPIF core function, between the API publishing function and the CAPIF core function and between the API management function and the CAPIF core function over CAPIF-3, CAPIF-4 and CAPIF-5 respectively can be API based.

The detailed information of the APIs provided by the CAPIF core function is specified in clause 10.

The security aspects of CAPIF reference points are specified in 3GPP TS 33.122 [12].

Figure 6.2.0-3 illustrates the CAPIF functional model using service-based interfaces.

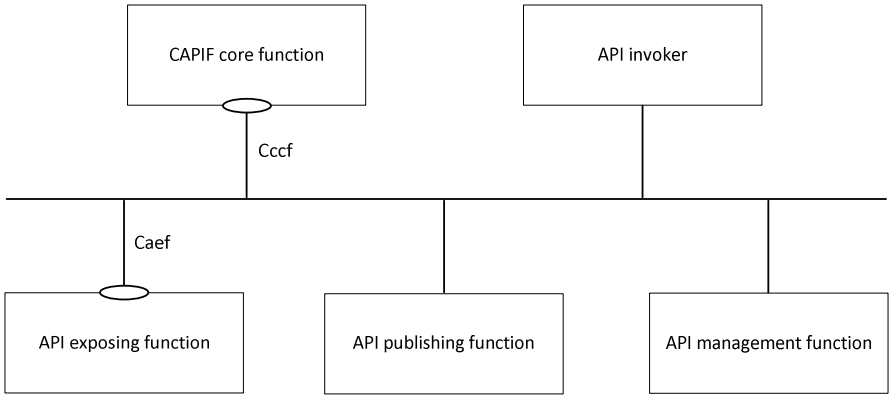

Figure 6.2.0-3: CAPIF functional model representation using service-based interfaces

Table 6.2.0-1 specifies the service-based interfaces supported by CAPIF.

Table 6.2.0-1: Service-based interfaces supported by CAPIF

Service-based interface	Entity	APIs offered
Cccf	CAPIF core function	Specified in subclause 10
Caef	API exposing function	Specified in subclause 11

6.2.1 Functional model description to support 3rd party API providers

Figure 6.2.1-1 shows the functional model for the CAPIF to support 3rd party API providers.

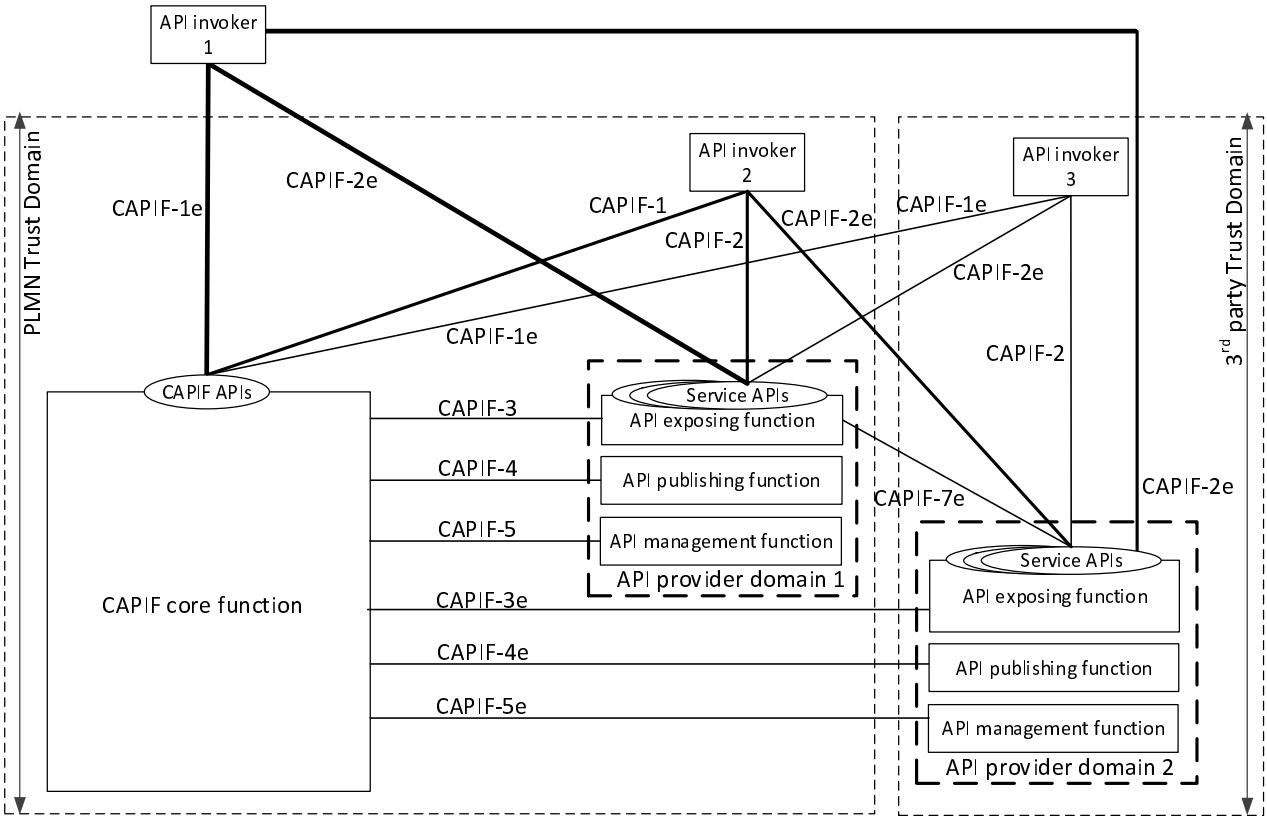

Figure 6.2.1-1: Functional model for the CAPIF to support 3rd party API providers

The CAPIF core function in the PLMN trust domain supports service APIs from both the PLMN trust domain and the 3rd party trust domain having business relationship with PLMN. The API invokers may exist within the PLMN trust domain, or within the 3rd party trust domain or outside of both the PLMN trust domain and the 3rd party trust domain. The API provider domain 1 offers the service APIs from the PLMN operator. The API provider domain 2 offers the service APIs from the 3rd party. When the 3rd party API provider is a trusted 3rd party of the PLMN, the API provider domain 1 also offers the service APIs from the 3rd party.

The API invoker 2 within the PLMN trust domain interacts with the CAPIF core function via CAPIF-1, and invokes the service APIs in the PLMN trust domain via CAPIF-2 and invokes the service APIs in the 3rd party trust domain via CAPIF-2e. The API invoker 3 within the 3rd party trust domain interacts with the CAPIF core function via CAPIF-1e, and invokes the service APIs in the PLMN trust domain via CAPIF-2e and invokes the service APIs in 3rd party trust domain via CAPIF-2. The API invoker 1 from outside the PLMN trust domain and 3rd party trust domain, interacts with the CAPIF core function via CAPIF-1e and invokes the service APIs in the PLMN trust domain and the service APIs in the 3rd party trust domain via CAPIF-2e.

The API exposing function, the API publishing function and the API management function of the API provider domain 1 within the PLMN trust domain interacts with the CAPIF core function via CAPIF-3, CAPIF-4 and CAPIF-5 respectively. The API exposing function, the API publishing function and the API management function of the API provider domain 2 within the 3rd party trust domain interacts with the CAPIF core function in the PLMN trust domain via CAPIF-3e, CAPIF-4e and CAPIF-5e respectively. The API exposing function within the PLMN trust domain and the 3rd party trust domain provides the service APIs to the API invoker, offered by the respective trust domains.

The interactions between the API exposing functions within the PLMN trust domain is via CAPIF-7 (not shown in the figure 6.2.1-1 for simplicity). The API exposing function within the PLMN trust domain interacts with the API exposing function in the 3rd party trust domain via CAPIF-7e.

NOTE 1: The communication between the API exposing function and the CAPIF core function, between the API publishing function and the CAPIF core function and between the API management function and the CAPIF core function over CAPIF-3/3e, CAPIF-4/4e and CAPIF-5/5e respectively can be API based.

The detailed information of the APIs provided by the CAPIF core function is specified in clause 10.

NOTE 2: The security aspects of CAPIF reference points are under SA3 responsibility and out of scope of the present document.

6.2.2 Functional model description to support CAPIF interconnection

Figure 6.2.2-1 shows the architectural model for the CAPIF interconnection which allows API invokers of a CAPIF provider to utilize the service APIs from the 3rd party CAPIF provider.

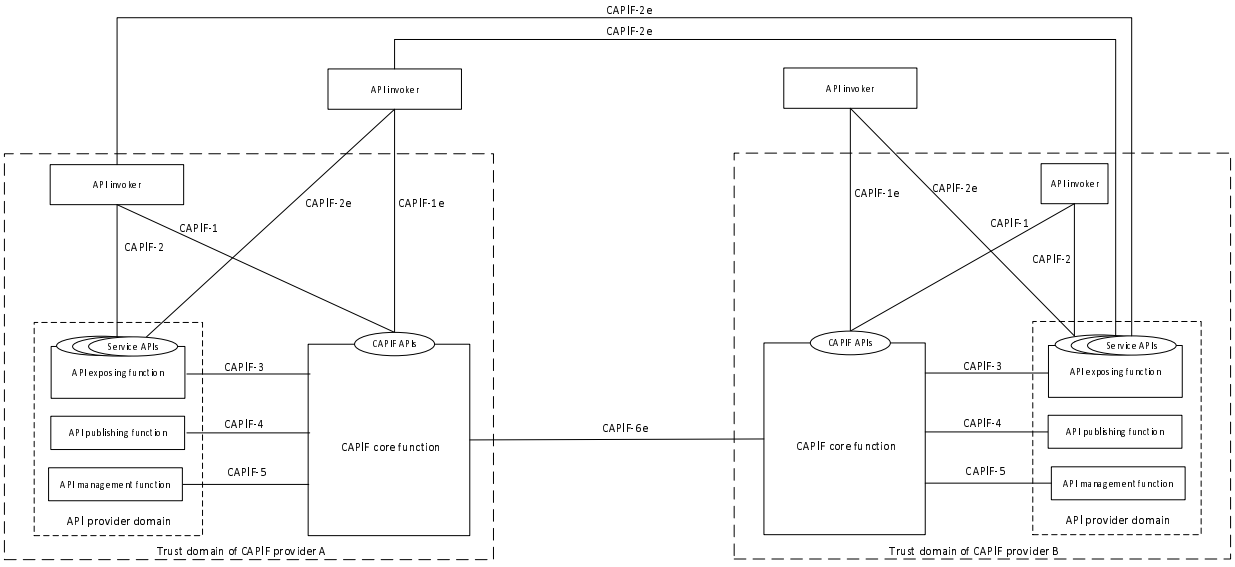

Figure 6.2.2-1: High level functional architecture for CAPIF interconnection with multiple CAPIF provider domains

Figure 6.2.2-2 shows the architectural model for the CAPIF interconnection within the same CAPIF provider domain, which allows API invokers of CAPIF core function 1 to utilize the service APIs from CAPIF core function 2, where both CAPIF core function 1 and CAPIF core function 2 are hosted within the trust domain of the CAPIF provider A.

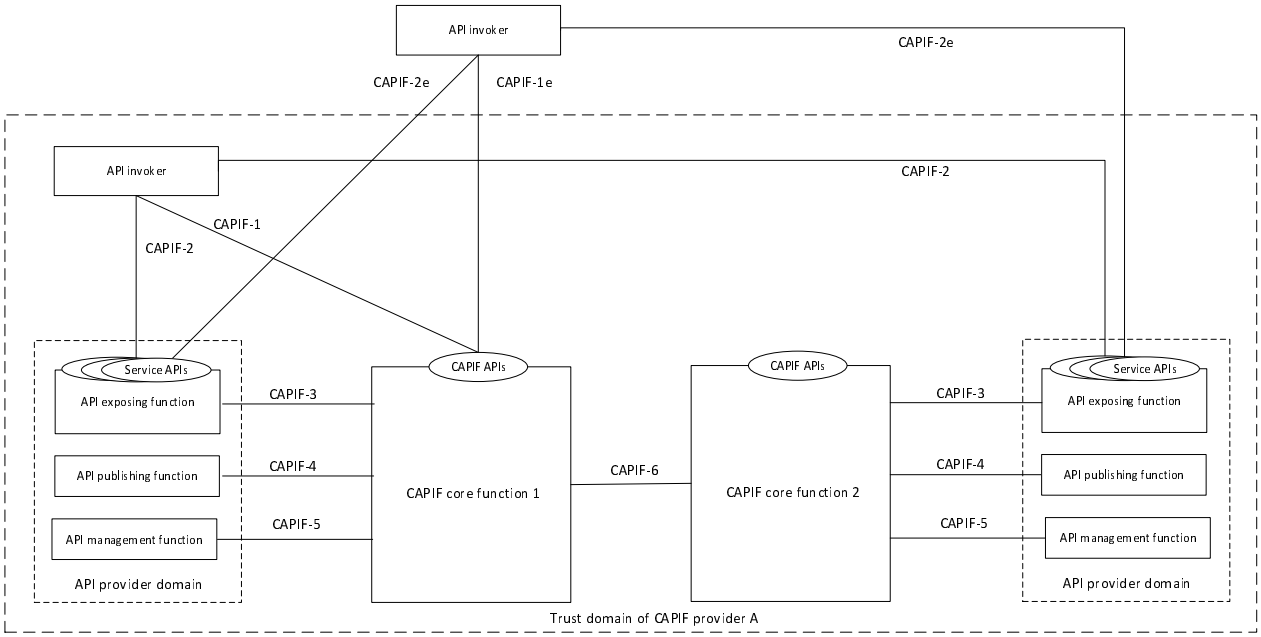

Figure 6.2.2-2: High level functional architecture for CAPIF interconnection within a CAPIF provider domain

The CAPIF provider A and CAPIF provider B host the CAPIF in their trust domains. A business relationship exists between the CAPIF providers.

The CAPIF providers in their respective trust domain hosts multiple CAPIF instances where each CAPIF instance consists of the CAPIF core function (local), the API provider domain and the API invokers. All interactions within the CAPIF instance is according to the functional model specified in clause 6.2.0.

When multiple CAPIF instances are deployed by a CAPIF provider there may be a hierarchy associated with the multiple CAPIF core function deployed which allows:

- the designated CAPIF core function of the CAPIF provider A to interconnect with the designated CAPIF core function of the CAPIF provider B; and
- within CAPIF provider A, one or more CAPIF core function interacts with the designated CAPIF core function 1.

The designated CAPIF core function of the CAPIF provider A provides the information about the CAPIF instances and service APIs deployed by the CAPIF provider A to the designated CAPIF core function of the CAPIF provider B and vice versa over CAPIF-6e reference point.

The CAPIF core function 2 of CAPIF provider A provides the information about the service APIs to the CAPIF core function 1 over CAPIF-6 reference point.

NOTE 1: Void

The API invokers may exist within the trust domain of CAPIF provider A, or within the trust domain of CAPIF provider B or outside of the trust domains of both CAPIF provider A and CAPIF provider B. The API invoker of a CAPIF provider is onboarded with the CAPIF core function in the corresponding trust domain of the CAPIF provider.

NOTE 2: For sake of simplicity, the service API interactions of API invokers of the CAPIF provider B are not shown. From each CAPIF provider's perspective the other CAPIF provider is a 3rd party.

One or more CAPIF core function can publish service APIs to the designated CAPIF core function over CAPIF-6 reference point and, also discover the service APIs from the designated CAPIF core function and vice versa over CAPIF-6 reference point.

The API invoker within the trust domain of CAPIF provider A interacts with the CAPIF core function of the CAPIF provider A via CAPIF-1 and discovers the service APIs of both CAPIF providers, and invokes the service APIs in the trust domain of CAPIF provider A via CAPIF-2 and invokes the service APIs in the trust domain of CAPIF provider B via CAPIF-2e. The API invoker from outside the trust domain of CAPIF providers, interacts with the CAPIF core function of th CAPIF provider A via CAPIF-1e and invokes the service APIs in the trust domain of the CAPIF providers via CAPIF-2e.

NOTE 3: The communication between the CAPIF core function of the CAPIF providers over CAPIF-6 or CAPIF-6e can be API based.

The detailed information of the APIs provided by the CAPIF core function is specified in clause 10.

NOTE 4: The security aspects of CAPIF reference points are under SA3 responsibility and out of scope of the present document.

NOTE 5: All interactions among entities within the CAPIF provider domains (regardless if CAPIF is deployed in a PLMN, SNPN or 3rd party network) are ruled by the functional model in clause 6.2.0, the support of 3rd party API providers is as in clause 6.2.1, whereas the interconnection among CCFs is according to this clause.

6.2.3 Functional model description to support RNAA

Figure 6.2.3-1 shows the architectural model for the RNAA which allows the resource owner to provide authorization to the API invocation.

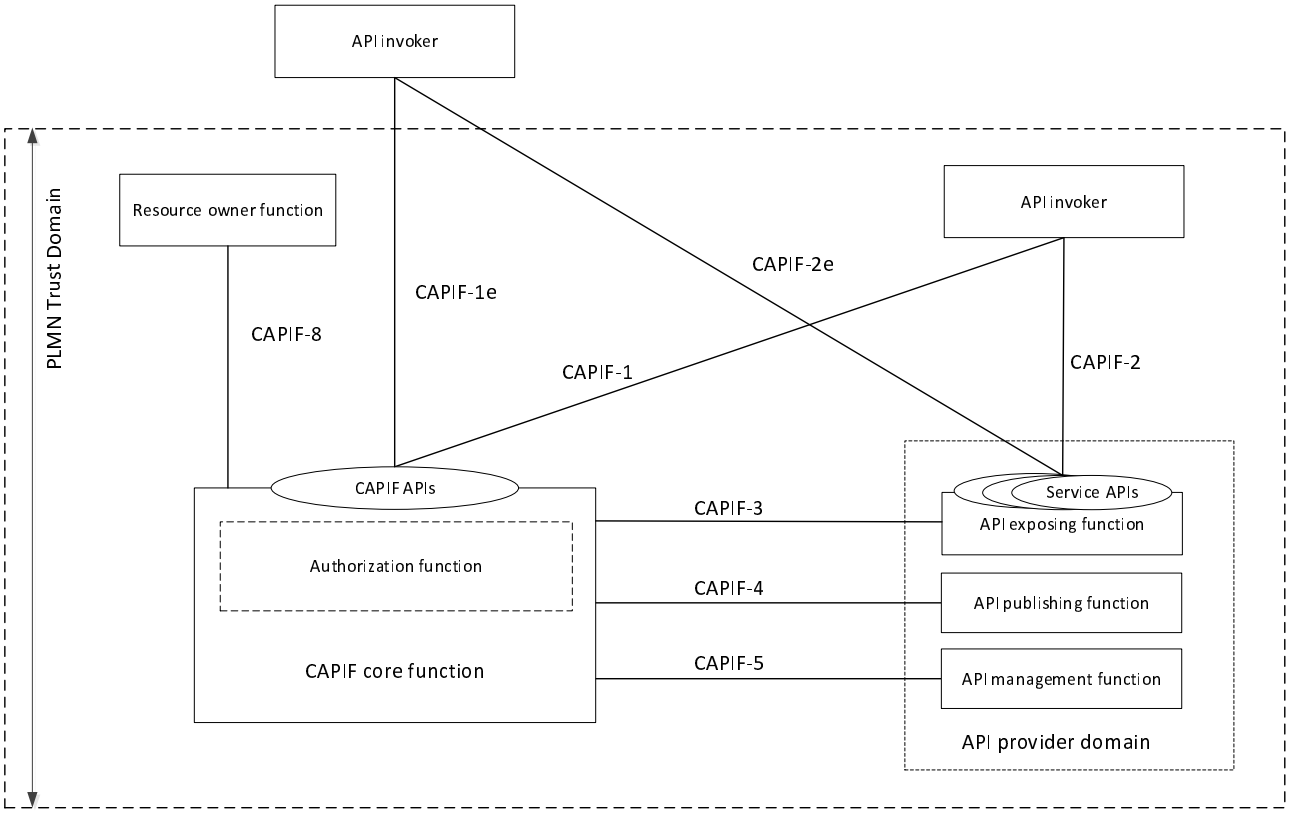

Figure 6.2.3-1: High level functional architecture for CAPIF supporting RNAA

The authorization function is an internal entity of the CAPIF core function.

The resource owner function interacts with the authorization function in the CAPIF core function via CAPIF-8. The resource owner function communicates with the authorization function in the CAPIF core function to manage resource owner consent. The CAPIF core function (authorization function) manages the consent information.

The API exposing function (e.g. NEF, SCEF) acts as a resource owner consent enforcement point and interacts with the authorization function in the CAPIF core function via CAPIF-3. The API exposing function can retrieve the resource owner consent parameters from the authorization function.

NOTE 1: RNAA is supported for both 4G and 5G network.

The API invoker interacts with authorization function in the CAPIF core function via CAPIF-1/CAPIF-1e.

NOTE 2: In the current release, 3rd party API providers (i.e., API providers outside the PLMN trust domain) are not supported for RNAA.

NOTE 3: The interaction between resource owner function and CCF over CAPIF-8 is not specified in the current release of the specification.

The security aspects of CAPIF supporting RNAA are specified in 3GPP TS 33.122 [12].

6.3 Functional entities description

6.3.1 General

Each subclause is a description of a functional entity and does not imply a physical entity.

6.3.2 API invoker

The API invoker is typically provided by a 3rd party application provider who has service agreement with PLMN operator. The API invoker may reside within the same trust domain as the PLMN operator network. The API invoker may be either an application on a server or an application on a UE.

The API invoker supports the following capabilities:

- Triggering API invoker onboarding/offboarding;
- Supporting the authentication by providing the API invoker identity and other information required for authentication of the API invoker;
- Supporting mutual authentication with CAPIF;
- Obtaining the authorization prior to accessing the service API;
- Discovering service APIs information; and
- Invoking the service APIs.

NOTE: The details of the specific service APIs are out of scope of the present document.

6.3.3 CAPIF core function

The CAPIF core function consists of the following capabilities:

- Authenticating the API invoker based on the identity and other information required for authentication of the API invoker;
- Supporting mutual authentication with the API invoker;
- Providing authorization for the API invoker prior to accessing the service API;
- Publishing, storing and supporting the discovery of service APIs information;
- Controlling the service API access based on PLMN operator configured policies;
- Storing the logs for the service API invocations and providing the service API invocation logs to authorized entities;

- Charging based on the logs of the service API invocations;
- Monitoring the service API invocations;
- Onboarding a new API invoker and offboarding an API invoker;
- Storing policy configurations related to CAPIF and service APIs;
- Support accessing the logs for auditing (e.g. detecting abuse); and
- Supports publishing, discovery of service APIs information with another CAPIF core function in CAPIF interconnection.

6.3.4 API exposing function

The API exposing function is the provider of the service APIs and is also the service communication entry point of the service API to the API invokers. The API exposing function consists of the following capabilities:

- Authenticating the API invoker based on the identity and other information required for authentication of the API invoker provided by the CAPIF core function;
- Validating the authorization provided by the CAPIF core function; and
- Logging the service API invocations at the CAPIF core function.

6.3.5 API publishing function

The API publishing function enables the API provider to publish the service APIs information in order to enable the discovery of service APIs by the API invoker. The API publishing function consists of the following capability:

- Publishing the service API information of the API provider to the CAPIF core function.

6.3.6 API management function

The API management function enables the API provider to perform administration of the service APIs. The API management function consists of the following capabilities:

- Auditing the service API invocation logs received from the CAPIF core function;
- Monitoring the events reported by the CAPIF core function;
- Configuring the API provider policies to the CAPIF core function;
- Monitoring the status of the service APIs;
- Onboarding the new API invokers and offboarding API invokers; and
- Registering and maintaining registration information of the API provider domain functions on the CAPIF core function.

NOTE: The API invoker onboarding/offboarding in the API management function is out of the scope of the current release.

6.3.7 Authorization function

The authorization function consists of the following capabilities:

- Receiving authorization from the resourse owner; and
- Providing the API invoker with the authorization information which is needed to access the resource owner’s resources.

NOTE: In the current release, the authorization function is an internal entity of the CAPIF core function.

6.3.8 Resource owner function

The resource owner function is responsible for interactions with the resource owner in a similar way to the resource owner's user agent shown in clause 4.1 of IETF RFC 6749 [13]. The resource owner function enables the following:

- Authorization for resource access; and
- Managing and revoking authorization for resource access.

NOTE: The procedures corresponding to these capabilities of the resource owner function are FFS and out of scope of the current release of the specification.

6.4 Reference points

6.4.1 General

The reference points for CAPIF are described in the following subclauses.

6.4.2 Reference point CAPIF-1 (between the API invoker and the CAPIF core function)

The CAPIF-1 reference point, which exists between the API invoker and the CAPIF core function, is used for the API invoker within the PLMN trust domain to discover service APIs, to authenticate and to get authorization.

The CAPIF-1 reference point supports:

- Onboarding the new API invokers and offboarding API invokers;
- Authenticating the API invoker based on the identity and credentials of the API invoker;
- Mutual authentication between the API invoker and the CAPIF core function;
- Providing authorization for the API invoker prior to accessing the service API;
- Providing authorization for the API invoker based on RNAA; and
- Discovering the service APIs information.

NOTE: The security aspects of CAPIF-1 are specified in subclause 6.2 of 3GPP TS 33.122 [12].

6.4.3 Reference point CAPIF-1e (between the API invoker and the CAPIF core function)

The CAPIF-1e reference point, which exists between the API invoker and the CAPIF core function, is used for the API invoker outside the PLMN trust domain to discover service APIs, to authenticate and to get authorization.

The CAPIF-1e reference point supports all the functions of CAPIF-1.

NOTE: The security aspects of CAPIF-1e are specified in subclause 6.3 of 3GPP TS 33.122 [12].

6.4.4 Reference point CAPIF-2 (between the API invoker and the API exposing function)

The CAPIF-2 reference point, which exists between the API invoker and the API exposing function belonging to the same trust domain, is used for the API invoker to communicate with the service APIs.

The CAPIF-2 reference point supports:

- Authenticating the API invoker based on the identity and credentials of the API invoker;
- Authorization verification for the API invoker upon accessing the service API; and

- Invocation of service APIs.

NOTE 1: The aspects related to the specific service API invocation in reference point CAPIF-2 are out of scope of the present document.

NOTE 2: The security aspects of CAPIF-2 are specified in subclause 6.4 of 3GPP TS 33.122 [12].

6.4.5 Reference point CAPIF-2e (between the API invoker and the API exposing function)

The CAPIF-2e reference point, which exists between the API invoker and the API exposing function belonging to a different trust domain, is used for the API invoker to communicate with the service APIs.

The CAPIF-2e reference point supports all the functions of CAPIF-2.

NOTE: The security aspects of CAPIF-2e are specified in subclause 6.5 of 3GPP TS 33.122 [12].

6.4.6 Reference point CAPIF-3 (between the API exposing function and the CAPIF core function)

The CAPIF-3 reference point, which exists between the API exposing function and the CAPIF core function, is used for exercising access and policy related control for service API communications initiated by the API invoker.

The CAPIF-3 reference point supports:

- Authenticating the API invoker based on the identity and credentials of the API invoker;
- Providing authorization for the API invoker prior to accessing the service API;
- Authorization verification for the API invoker upon accessing the service API;
- Authorization verification for the API invoker based on RNAA;
- Controlling the service API access based on PLMN operator configured policies;
- Logging the service API invocations; and
- Charging the service API invocations.

NOTE: The security aspects of CAPIF-3 are specified in subclause 6.6 of 3GPP TS 33.122 [12].

6.4.7 Reference point CAPIF-4 (between the API publishing function and the CAPIF core function)

The CAPIF-4 reference point, which exists between the API publishing function and the CAPIF core function, is used for publishing the service API information.

The CAPIF-4 reference point supports:

- Publishing the service APIs information by the API publishing function.

NOTE: The security aspects of CAPIF-4 are specified in subclause 6.6 of 3GPP TS 33.122 [12].

6.4.8 Reference point CAPIF-5 (between the API management function and the CAPIF core function)

The CAPIF-5 reference point, which exists between the API management function and the CAPIF core function, is used for management of service API, API invoker and API provider domain function information.

The CAPIF-5 reference point supports:

- Accessing the service API invocation logs by the API management function;

- Enabling the API management function to monitor the events reported due to the service APIs invocations;
- Onboarding new API invokers by provisioning the API invoker information at the CAPIF core function, requesting explicit grant of new API invokers onboarding and confirming onboarding success;
- Offboarding API invokers;
- Enabling the API management function to configure policies at the CAPIF core function e.g. service API invocation throttling, blocking API invocation for certain duration;
- Enabling the API provider to monitor the status of service APIs (e.g. pilot or live status, start or stop status of service API);
- Registering API provider domain functions on the CAPIF core function; and
- Update of the registration information of API provider domain functions on the CAPIF core function.

NOTE 1: The security aspects of CAPIF-5 are specified in subclause 6.6 of 3GPP TS 33.122 [12].

NOTE 2: The API invoker onboarding/offboarding over CAPIF-5 is out of the scope of the current release.

6.4.9 Reference point CAPIF-3e (between the API exposing function and the CAPIF core function)

The CAPIF-3e reference point, which exists between the API exposing function within the 3rd party trust domain and the CAPIF core function within the PLMN trust domain, is used for exercising access and policy related control for service API communications initiated by the API invoker.

The CAPIF-3e supports all the functions of CAPIF-3.

NOTE: The security aspects of CAPIF-3e are specified in clause 6.10 of 3GPP TS 33.122 [12].

6.4.10 Reference point CAPIF-4e (between the API publishing function and the CAPIF core function)

The CAPIF-4e reference point, which exists between the API publishing function within the 3rd party trust domain and the CAPIF core function within the PLMN trust domain, is used for publishing the service API information.

The CAPIF-4e reference point supports all the functions of CAPIF-4.

NOTE: The security aspects of CAPIF-4e are specified in clause 6.10 of 3GPP TS 33.122 [12].

6.4.11 Reference point CAPIF-5e (between the API management function and the CAPIF core function)

The CAPIF-5e reference point, which exists between the API management function within the 3rd party trust domain and the CAPIF core function within the PLMN trust domain, is used for management of service API, API invoker and API provider domain function information.

The CAPIF-5e reference point supports all the functions of CAPIF-5.

NOTE: The security aspects of CAPIF-5e are specified in clause 6.10 of 3GPP TS 33.122 [12].

6.4.12 Reference point CAPIF-7 (between the API exposing functions)

The CAPIF-7 reference point, which exists between the API exposing functions belonging to the same trust domain, is used for the forwarding or routing of the API invoker's service API invocation from one API exposing function to the other API exposing function deployed in the PLMN trust domain.

The CAPIF-7 reference point supports all the functions of CAPIF-2.

The CAPIF-7 reference point supports invocation of service APIs originated by the API invoker using CAPIF-2.

NOTE 1: The aspects related to the specific service API invocation in reference point CAPIF-7 are out of scope of the present document.

NOTE 2: The security aspects of CAPIF-7 are the responsibility of SA3.

6.4.13 Reference point CAPIF-7e (between the API exposing functions)

The CAPIF-7e reference point, which exists between the API exposing functions belonging to different trust domains, is used for the forwarding or routing of the API invoker's service API invocation from one API exposing function to the other API exposing function between different trust domains.

The CAPIF-7e reference point supports all the functions of CAPIF-2e.

NOTE: The security aspects of CAPIF-7e are the responsibility of SA3.

6.4.14 Reference point CAPIF-6 (between the CAPIF core functions of the same CAPIF provider)

The CAPIF-6 reference point exists between the CAPIF core functions within the same trust domain of CAPIF provider.

The CAPIF-6 reference point supports:

- Publishing the service APIs information; and
- Discovering the service APIs information.

6.4.15 Reference point CAPIF-6e (between the CAPIF core functions of different CAPIF providers)

The CAPIF-6e reference point exists between the CAPIF core function within the 3rd party trust domain and the CAPIF core function within the PLMN trust domain.

The CAPIF-6e reference point supports all the functions of CAPIF-6.

NOTE: The security aspects of CAPIF-6e will be specified by SA3.

Editor's note: Reference to the appropriate SA3 specification is needed.

6.4.16 Reference point CAPIF-8 (between the CAPIF core function and the resource owner function)

The CAPIF-8 reference point exists between the CAPIF core function and the resource owner function.

The CAPIF-8 reference point supports:

- Providing authorization for resource access; and
- Managing and revoking the provided authorization.

NOTE: The functionalities over CAPIF-8 is FFS and out of scope of the current release of the specification.

6.5 Service-based interfaces

The CAPIF architecture contains the following service-based interfaces:

- Cccf: Service-based interface exhibited by CAPIF core function.
- Caef: Service-based interface exhibited by API exposing function.

7 Application of functional model to deployments

7.1 General

The CAPIF deployments in centralized and distributed models are described in clause 7.2 and clause 7.3. The multiple CCFs deployment is described in clause 7.4.

The RNAA deployments are described in clause 7.5.

The CAPIF deployment models shown are not exhaustive.

7.2 Centralized deployment

The CAPIF can be deployed centrally as illustrated in the figure 7.2-1.

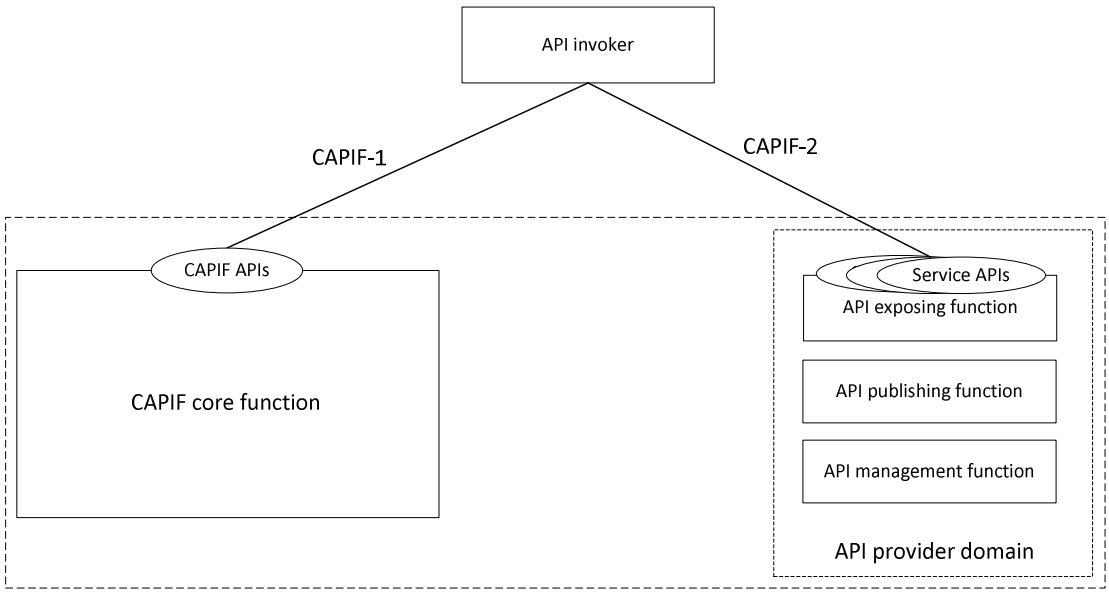

Figure 7.2-1: Centralized deployment of CAPIF

In one centralized deployment, the CAPIF core function and the API provider domain functions are co-located. The API invoker can interact independently with the CAPIF core function and the API exposing function including the service APIs. The CAPIF appears as a gateway for all API invoker interactions. The API invoker obtains the service API information and its entry point details from the CAPIF core function via CAPIF-1. The service communication point of entry for the service API is the API exposing function which also applies any access control or policy control to the internal interactions between the API invoker and the service API in coordination with the CAPIF core function.

NOTE: The API invoker can be outside the PLMN trust domain and will access the CAPIF via CAPIF-1e and CAPIF-2e instead of CAPIF-1 and CAPIF-2.

Another variation of the centralized deployment is where the CAPIF core function and the API exposing function is co-located where as other API provider domain functions (API publishing function and API management function) are not co-located with the API exposing function. In such deployment scenario, the CAPIF core function interacts with the API publishing function and the API management function via CAPIF-4 and CAPIF-5 reference points respectively.

7.3 Distributed deployment

The CAPIF can be deployed in a distributed manner illustrated in the figure 7.3-1.

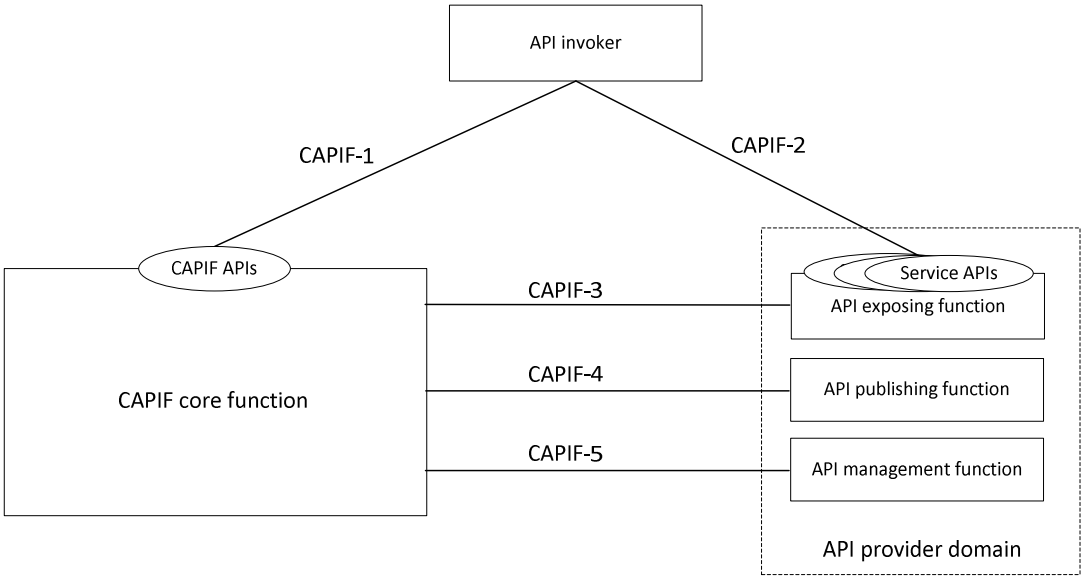

Figure 7.3-1: Distributed deployment of the CAPIF within PLMN trust domain

In distributed deployment, the CAPIF core function and the API provider domain functions are not co-located. The CAPIF core function interacts with the API exposing function, the API publishing function and the API management function via CAPIF-3, CAPIF-4 and CAPIF-5 reference points respectively. The API invoker can interact independently with the CAPIF core function and the API exposing function including the service APIs. In this deployment, the API exposing function appears as an agent for all service API invocations from the API invoker. The API invoker obtains the service API information and its entry point details from the CAPIF core function via CAPIF-1 interface. The first point of entry for the service API is the API exposing function during API invocation. The API exposing function acts as agent for service API applying any access control or policy control to the interactions between the API invoker and the service API in coordination with the CAPIF core function via CAPIF-3 interface.

The CAPIF can be deployed by splitting the functionality of the API exposing function among multiple API exposing function entities, of which one acts as the entry point. However there will be single API publishing function and single API management function in the API provider domain although there could be multiple API exposing function entities. The CAPIF deployment with cascading API exposing functions is as illustrated in the figure 7.3-2.

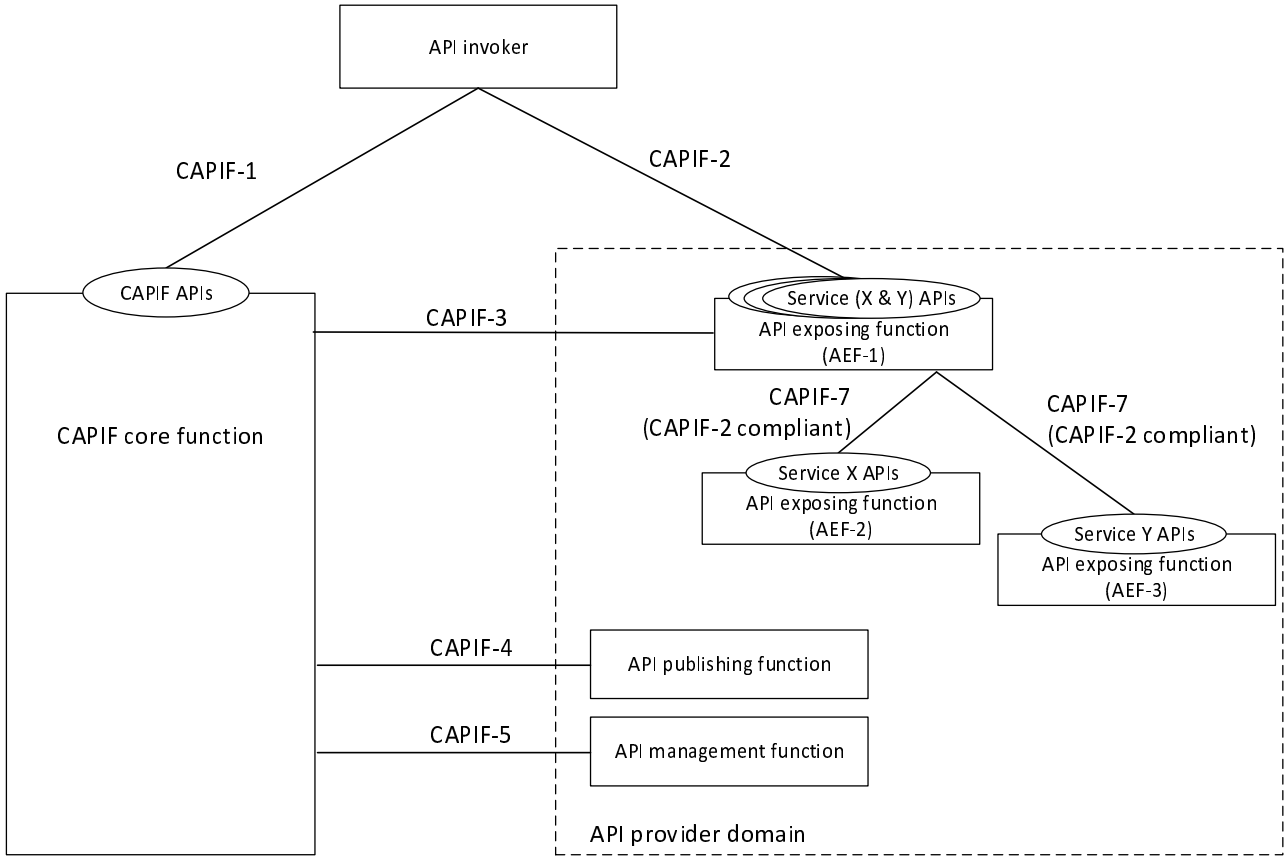

Figure 7.3-2: Distributed deployment of the CAPIF with cascading API exposing functions

In this deployment option, the API exposing function can have several instances like AEF-1, AEF-2 and AEF-3 which can be assigned with different roles. The roles for each API exposing function are decided by the operator. In this illustration, the API exposing functions AEF-2 and AEF-3 provide service APIs for service X and service Y respectively. The API exposing function AEF-1 provides the service communication entry point to the service APIs for service X APIs and service Y APIs. The API exposing function AEF-1 for instance can hide the topology of service X APIs and service Y APIs from the API invoker. The API exposing function AEF-1 also applies any access control or policy control to the interactions between the API invoker and service X APIs and between the API invoker and service Y APIs, in coordination with the CAPIF core function using CAPIF-3.

The API invoker interacts with the CAPIF core function via CAPIF-1. The API invoker interacts with service (X&Y) APIs on the API exposing function AEF-1 via CAPIF-2. The API exposing function AEF-1 forwards the invocation of the service X API or service Y API from the API invoker to the API exposing functions AEF-2 or AEF-3 respectively via CAPIF-2. The API messages are forwarded via CAPIF-7 (in compliance with CAPIF-2 interaction between the API invoker and the AEF-1) in the interactions between API exposing functions. The API invoker cannot directly interact with service X APIs and service Y APIs provided by API exposing functions AEF-2 and AEF-3 respectively.

Different splits of responsibility are possible. In another example illustrated in figure 7.3-3, the API exposing function AEF-1 could provide topology hiding for API exposing functions AEF-2 and AEF-3, plus access control for AEF-3. The API exposing function AEF-2 would provide its own access control, interacting with the CAPIF core function via CAPIF-3.

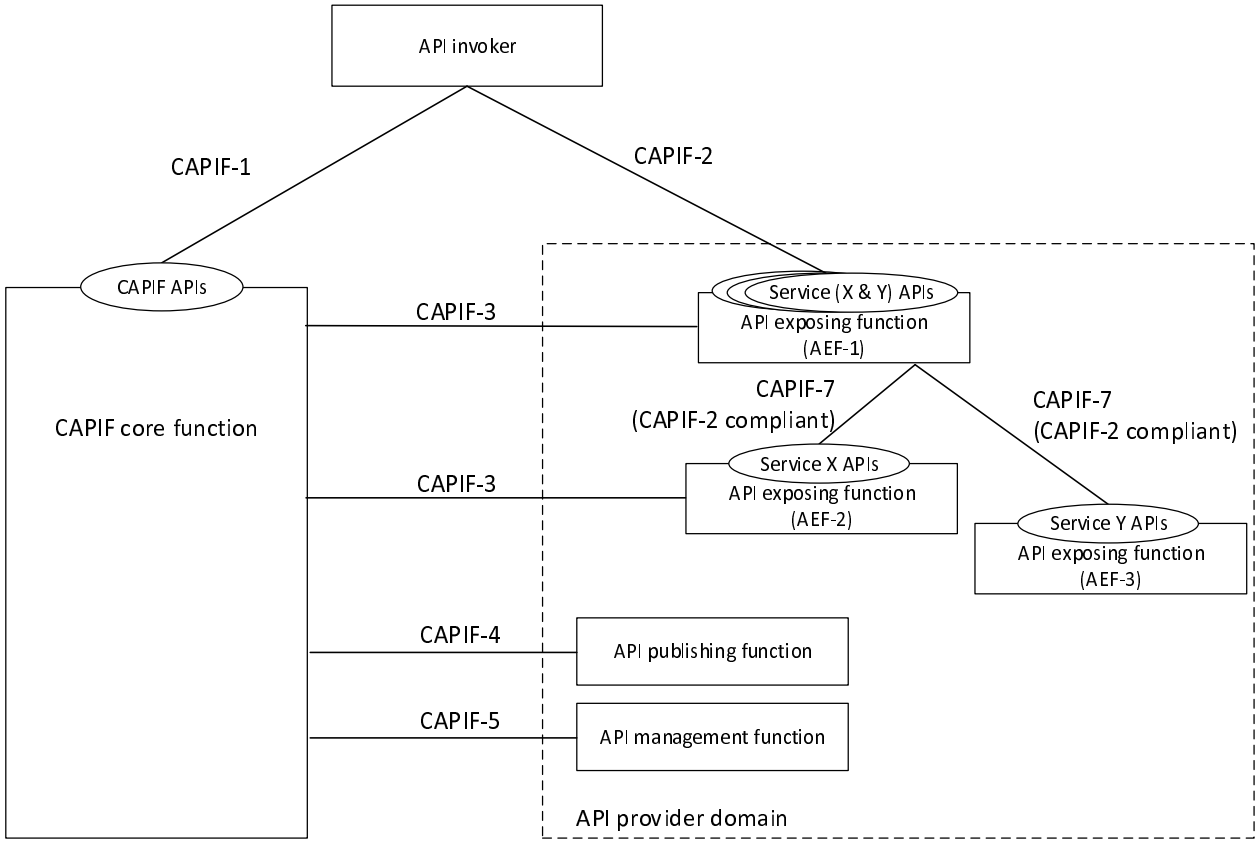

Figure 7.3-3: Another example of distributed deployment of the CAPIF with cascading API exposing functions

NOTE 1: The API invoker can be outside the PLMN trust domain and will access the CAPIF via CAPIF-1e and CAPIF-2e instead of CAPIF-1 and CAPIF-2.

When considering the 3rd party trust domain deployment, the API provider domain belongs to a 3rd party trust domain, the CAPIF core function belongs to PLMN trust domain and the API invoker belongs to PLMN trust domain as illustrated in figure 7.3-4.

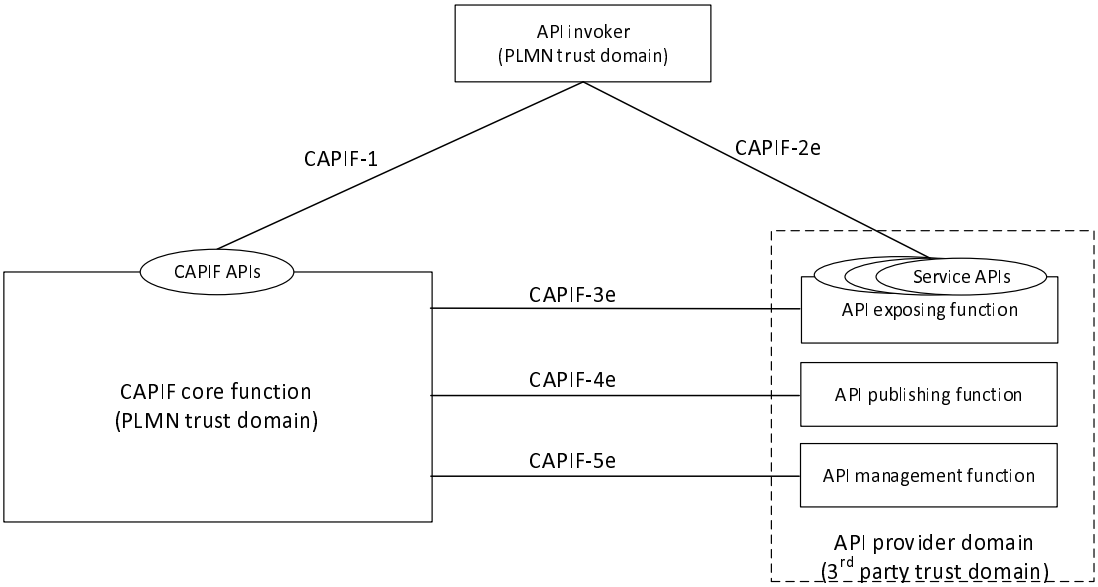

Figure 7.3-4: Distributed deployment of CAPIF considering PLMN trust domain and 3rd party trust domain

The interactions between the AEF and the CAPIF core function is based on CAPIF-3e. The interactions between the API publisher function and the CAPIF core function is based on CAPIF-4e. The interactions between the API management function and the CAPIF core functions are based on CAPIF-5e. The interactions between the API invoker and the AEF are based on CAPIF-2e. The API provider domain functions may be deployed in the PLMN trust domain and the interactions of the API provider domain functions within CAPIF of the PLMN trust domain is not shown in the figure 7.3-4 and is as illustrated in figure 7.3-1.

NOTE 2: For deployments illustrated in figure 7.3-2 and figure 7.3-3, when the API provider domain belongs to the 3rd party trust domain, the interactions between the AEF of the API provider domain and API invoker belonging to the PLMN trust domain are carried over CAPIF-2e reference point and the interactions between the entities of the API provider domain and the CAPIF core function belonging to the PLMN trust domain are carried over CAPIF-3e, CAPIF-4e and CAPIF-5e as illustrated in figure 7.3-4.

7.4 Multiple CCFs deployment

Multiple CAPIF core functions may be deployed within the PLMN trust domain as illustrated in the figure 7.4-1. For simplicity, the API invoker is not shown.

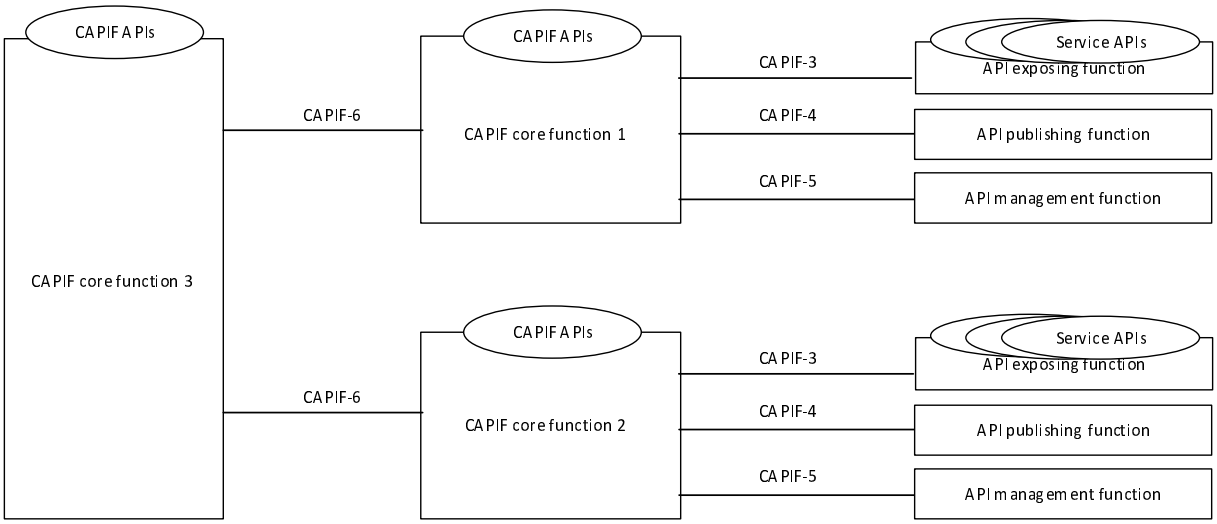

Figure 7.4-1: Multiple CCFs deployment within the PLMN trust domain

In the distributed deployment, the CAPIF core function 1 and the CAPIF core function 2 interact with CAPIF core function 3 via CAPIF-6 reference point. The CAPIF core function 3 assumes the role of a centralized repository of service APIs in the PLMN trust domain.

NOTE: The CAPIF core function 3 can be connected with the API exposing function(s) and API invokers.

The CAPIF core function 1 and the CAPIF core function 2 publishes the service API provided by its connected API exposing function(s) to the CAPIF core function 3, and obtains the service API information provided by other CAPIF core function(s).

An API invoker (not shown in the figure for simplicity) connected to the CAPIF core function 1 is able to discover and invoke the service APIs provided by the API exposing function connected to the CAPIF core function 2.

7.5 RNAA deployments

CAPIF supports RNAA and has enabled API invoker(s) to have authorized access to resources of a resource owner provided by service APIs offered by the AEF. The CCF acts as the Authorization Function and supports the authentication and authorization of the resource owner. Based on resource owner's authorization, the CCF provides the access token for a service API access to the API invoker. The API invoker performs service API invocations on the AEF by utilizing the access token.

The API invoker may be deployed in the following ways:

a. API invoker may be deployed as AF on the UE (i.e. 3rd party application).

b. API invoker may be deployed as AF on the UE supporting several other 3rd party applications deployed on the UE.

c. API invoker may be deployed on the network as AF.

The resource owner is connected via a UE and can use a Resource Owner Function deployed on the UE to interact with the CCF acting as the Authorization Function for authentication and authorization i.e., granting permission to the API invoker to access resource(s) of the resource owner provided by the service API.

NOTE: The details of the protocol for CAPIF supporting RNAA is specified in 3GPP TS 33.122 [12].

When API invoker is deployed on a UE (cases a and b), the API invoker is allowed to access the resources of the resource owner corresponding to the UE.

8 Procedures and information flows

8.1 Onboarding the API invoker to the CAPIF

8.1.1 General

The procedure in this subclause corresponds to the architectural requirements for onboarding the API invoker to the CAPIF. The CAPIF enables a one time onboarding process that enrolls the API invoker as a recognized user of the CAPIF, which may be triggered by the API invoker via CAPIF-1 or CAPIF-1e, or may be based on provisioning.

8.1.2 Information flows

8.1.2.1 Onboard API invoker request

Table 8.1.2.1-1 describes the information flow onboard API invoker request from the API invoker to the CAPIF core function.

Table 8.1.2.1-1: Onboard API invoker request

Information element	Status	Description
Onboarding information	M	The information of the API invoker including enrolment details, required for onboarding
APIs for enrollment	O	List of APIs being enrolled for.
Proposed expiration time	O	Proposed expiration time for the onboarding.

8.1.2.2 Onboard API invoker response

Table 8.1.2.2-1 describes the information flow onboard API invoker response from the CAPIF core function to the API invoker.

Table 8.1.2.2-1: Onboard API invoker response

Information element	Status	Description
Onboarding status	M	The result of onboarding request i.e., success indication is included if the API invoker is granted permission otherwise failure.
Enrolled information	O (see NOTE 1)	Information from the provisioned API invoker profile which may include information to allow the API invoker to be authenticated and to obtain authorization for service APIs
Service API information	O (see NOTE 2)	The service API information as specified in Table 8.7.2.2-1.
Reason	O (see NOTE 3)	This element indicates the reason when onboarding status is failure.
Expiration time	O	Indicates the expiration time of the onboarding. At expiration, CCF cancels the enrollment of the API invoker from CAPIF. If omitted, it indicates the onboarding does not expire.
NOTE 1: Information element shall be present when onboarding status is successful. NOTE 2: Information element may be present when onboarding status is successful. NOTE 3: Information element shall be present when onboarding status is failure.		

8.1.3 Procedure

Figure 8.1.3-1 illustrates the procedure for onboarding the API invoker to the CAPIF. The security aspects of this procedure are specified in subclause 6.1 of 3GPP TS 33.122 [12].

Pre-conditions:

1. The API invoker is not a recognized user of the CAPIF.
2. The API invoker has visibility to APIs information (e.g., API catalogue or dashboard - central place for the API provider to manage which APIs are displayed, giving API invokers the ability to enroll for).

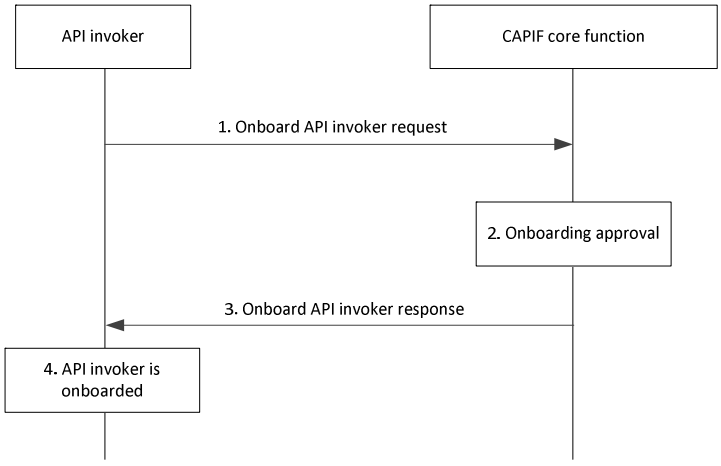

Figure 8.1.3-1: Procedure for onboarding the API invoker to the CAPIF

1. For enrollment of the API invoker to be a recognized user of the CAPIF, the API invoker triggers onboard API invoker request towards the CAPIF core function, providing the information as required for the API management.
2. The CAPIF core function begins the onboarding process by verifying whether all the necessary information has been provided to onboard the API invoker, and further initiates a grant process. Successful onboarding results in provisioning API invoker profile which includes identity for the API invoker. The authorization information and the list of APIs and the categories of APIs that the API invoker can access subsequent to successful onboarding may also be created.

NOTE 1: Completion of onboarding process can require explicit grant by the CAPIF administrator or the API management, which is left out-of-scope of this solution. CAPIF can handle the grant process internally without the need of explicit grant by the CAPIF administrator.

NOTE 2: The API invoker profile consists of at least the identity information for the API invoker, information required for the authentication and authorization by the CAPIF and the CAPIF identity information.

3. If the API invoker has triggered the onboard API invoker request and is granted permission, the onboard API invoker response provides success indication including information from the provisioned API invoker profile which may include information to allow the API invoker to be authenticated and to obtain authorization for service APIs.

4. As a result of successful onboarding process, the CAPIF core function is able to authenticate and authorize the API invoker.

8.2 Offboarding the API invoker from the CAPIF

8.2.1 General

This subclause defines the procedure for offboarding the API invoker from the CAPIF. The offboarding process makes the API invoker no longer a recognized user of the CAPIF. The procedure is triggered by the API invoker over CAPIF-1 or CAPIF-1e.

8.2.2 Information flows

This subclause describes the information flows for the API invoker offboarding.

8.2.2.1 Offboard API invoker request

Table 8.2.2.1-1 describes the information flow offboard API invoker request from the API invoker to the CAPIF core function.

Table 8.2.2.1-1: Offboard API invoker request

Information element	Status	Description
API invoker identity information	M	Identity information of the API invoker requesting offboarding
Reason	O	Indicate the reason of offboarding

8.2.2.2 Offboard API invoker response

Table 8.2.2.2-1 describes the information flow offboard API invoker response from the CAPIF core function to the API invoker.

Table 8.2.2.2-1: Offboard API invoker response

Information element	Status	Description
Result	M	Indicates the success or failure of the offboarding operation

8.2.3 Procedure

Figure 8.2.3-1 illustrates the procedure for offboarding the API invoker from the CAPIF, triggered by the API invoker. The security aspects of this procedure are specified in subclause 6.8 of 3GPP TS 33.122 [12].

Pre-conditions:

1. The API invoker has been onboarded as a recognized user of the CAPIF.

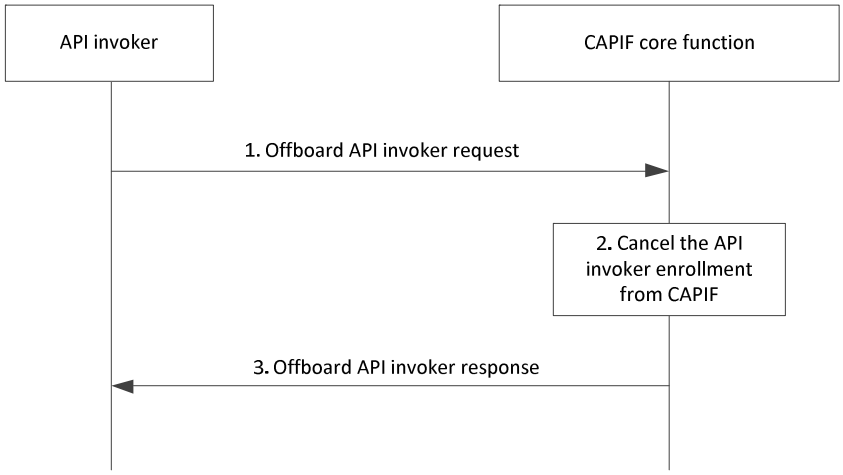

Figure 8.2.3-1: Procedure for offboarding the API invoker from the CAPIF

1. The API invoker triggers offboard API invoker request to the CAPIF core function, providing the information as required for the API management.

2. The CAPIF core function cancels the enrollment of the API invoker from CAPIF. The API invoker ceases to be a recognized user of the CAPIF. All the authorizations corresponding to the API invoker are revoked from CAPIF. Optionally, the information of the API invoker may be retained at the CAPIF core function as per the operator policy.

NOTE: Completion of offboarding process can require explicit notification to the CAPIF administrator or the API management, which is left out-of-scope of this solution. CAPIF can handle the de-provisioning process internally without the need of explicit grant by the CAPIF administrator.

3. The CAPIF core function returns the offboard API invoker response providing successful offboarding indication.

8.3 Publish service APIs

8.3.1 General

The CAPIF supports publishing service APIs by the API provider. The API publishing function can be within PLMN trust domain or within 3rd party trust domain.

8.3.2 Information flows

8.3.2.1 Service API publish request

Table 8.3.2.1-1 describes the information flow service API publish request from the API publishing function to the CAPIF core function.

Table 8.3.2.1-1: Service API publish request

Information element	Status	Description
API publisher information	M	The information of the API publisher may include identity, authentication and authorization information
Service API information	M	The service API information includes the service API name, API provider name (optional), List of public IP ranges of UEs (optional), service API category, service API status (e.g. active, inactive), communication type, description, Serving Area Information (optional), AEF location (optional), interface details (e.g. IP address, port number, URI), protocols, version numbers, and data format, Service KPIs (optional).
Shareable information	O (see NOTE)	Indicates whether the service API or the service API category can be published to other CCFs. And if sharing, a list of CAPIF provider domain information where the service API or the service API category can be published is contained.
NOTE: If the shareable information is not present, the service API is not allowed to be shared.		

The Service KPIs is defined as below:

Table 8.3.2.1-2: Service KPIs

Information element	Status	Description
Maximum Request rate	O	Maximum request rate from the API Invoker supported by the server.
Maximum Response time	O	The maximum response time advertised for the API Invoker's service requests.
Availability	O	Advertised percentage of time the server is available for the API Invoker's use.
Available Compute	O	The maximum compute resource available for the API Invoker.
Available Graphical Compute	O	The maximum graphical compute resource available for the API Invoker.
Available Memory	O	The maximum memory resource available for the API Invoker.
Available Storage	O	The maximum storage resource available for the API Invoker.
Connection Bandwidth	O	The connection bandwidth in Kbit/s advertised for the API Invoker's use.

8.3.2.2 Service API publish response

Table 8.3.2.2-1 describes the information flow service API publish response from the CAPIF core function to the API publishing function.

Table 8.3.2.2-1: Service API publish response

Information element	Status	Description
Result	M	Indicates the success or failure of publishing the service API information
Service API published information reference	O (see NOTE)	The information which can be used for referencing the information (set) about the published service API by the API publishing function.
Service API information	O (see NOTE)	The authorized service API information.
NOTE: This information element is included when the Result indicates success.		

8.3.3 Procedure

Figure 8.3.3-1 illustrates the procedure for publishing the service APIs. The service API publish mechanism is supported by the CAPIF core function.

Pre-conditions:

1. Authorization details of the APF are available with the CAPIF core function.
2. API invokers may have subscribed with the CAPIF core function to obtain new service API information.

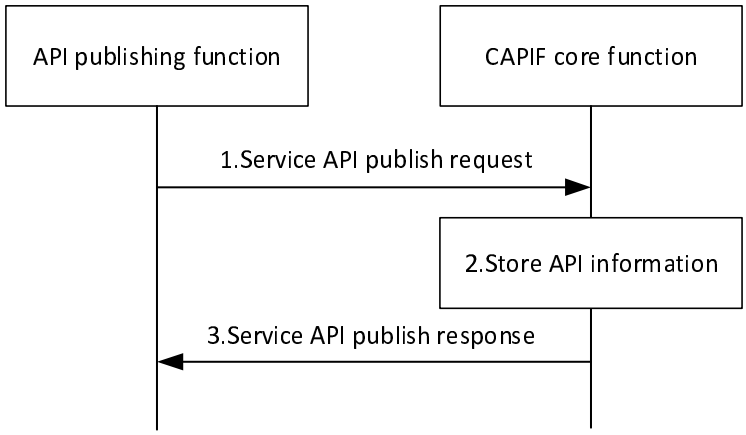

Figure 8.3.3-1: Publish service APIs

1. The API publishing function sends a service API publish request to the CAPIF core function, with the details of the service API. If the service API is to be shared to other CAPIF core functions, the shareable information and the CAPIF provider domain information are included.
2. Upon receiving the service API publish request, the CAPIF core function checks whether the API publishing function is authorized to publish service APIs. If the check is successful, the service API information provided by the API publishing function is stored at the CAPIF core function (API registry).
3. The CAPIF core function provides a service API publish response to the API publishing function indicating success or failure result and triggers notifications to subscribed API invokers as described in subclause 8.8.4.

8.4 Unpublish service APIs

8.4.1 General

The CAPIF supports unpublishing service APIs by the API provider. Once the service API information is unpublished, it is no more available to be discovered by API invokers. The API publishing function can be within PLMN trust domain or within 3rd party trust domain.

8.4.2 Information flows

8.4.2.1 Service API unpublish request

Table 8.4.2.1-1 describes the information flow service API unpublish request from the API publishing function to the CAPIF core function.

Table 8.4.2.1-1: Service API unpublish request

Information element	Status	Description
API publisher information	M	The information of the API publisher may include identity, authentication and authorization information
Service API published information reference	M	The information provided by the CAPIF core function which can be for referencing the information (set) about the published service API by the API publishing function.

8.4.2.2 Service API unpublish response

Table 8.4.2.2-1 describes the information flow service API unpublish response from the CAPIF core function to the API publishing function.

Table 8.4.2.2-1: Service API unpublish response

Information element	Status	Description
Result	M	Indicates the success or failure of unpublishing the service API information

8.4.3 Procedure

Figure 8.4.3-1 illustrates the procedure for unpublishing the service APIs. The service API unpublish mechanism is supported by the CAPIF core function.

Pre-conditions:

1. Authorization details of the APF are available with the CAPIF core function.
2. API invokers may have subscribed with the CAPIF core function to obtain notification regarding service API unpublish.

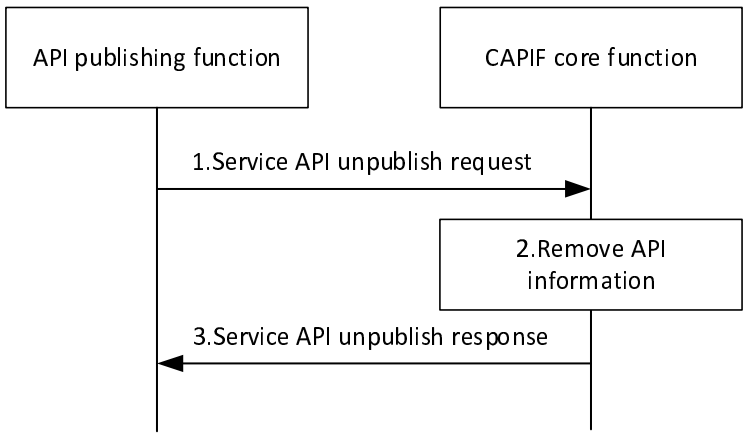

Figure 8.4.3-1: Unpublish service APIs

1. The API publishing function sends a service API unpublish request to the CAPIF core function, with service API published information reference provided by the CAPIF core function when the service API was published.
2. Upon receiving the service API unpublish request, the CAPIF core function checks whether the API publishing function is authorized to unpublish service APIs. If the check is successful, the service API information provided by the API publishing function is removed at the CAPIF core function (API registry).
3. The CAPIF core function provides a service API unpublish response to the API publishing function and triggers notifications to subscribed API invokers as described in subclause 8.8.4.

8.5 Retrieve service APIs

8.5.1 General

The CAPIF supports retrieving the published service APIs information by the API provider. The API publishing function can be within PLMN trust domain or within 3rd party trust domain.

8.5.2 Information flows

8.5.2.1 Service API get request

Table 8.5.2.1-1 describes the information flow service API get request from the API publishing function to the CAPIF core function.

Table 8.5.2.1-1: Service API get request

Information element	Status	Description
API publisher information	M	The information of the API publisher may include identity, authentication and authorization information
Service API published information reference	M	The information provided by the CAPIF core function which can be for referencing the information (set) about the published service API by the API publishing function.

8.5.2.2 Service API get response

Table 8.5.2.2-1 describes the information flow service API get response from the CAPIF core function to the API publishing function.

Table 8.5.2.2-1: Service API get response

Information element	Status	Description
Result	M	Indicates the success or failure of retrieving the service API information
Service API information	O (see NOTE)	The service API information as specified in Table 8.3.2.1-1.
NOTE: Shall be present if the Result information element indicates that the service API get request is successful. Otherwise service API information shall not be present.		

8.5.3 Procedure

Figure 8.5.3-1 illustrates the procedure for retrieving the service APIs. The service API retrieval mechanism is supported by the CAPIF core function.

Pre-condition:

1. Authorization details of the APF are available with the CAPIF core function.

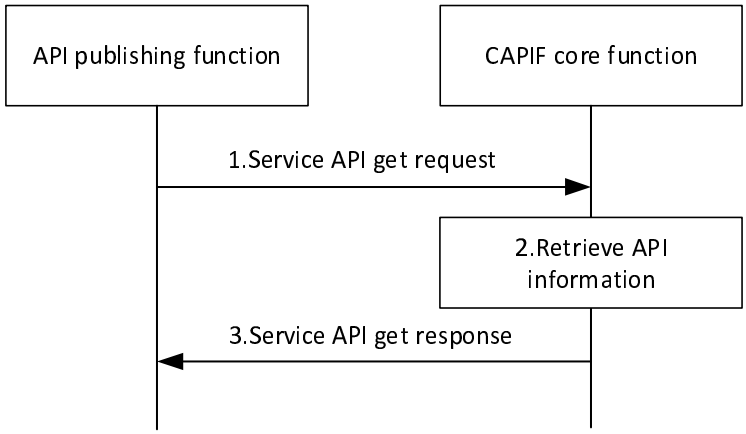

Figure 8.5.3-1: Retrieve service APIs

1. The API publishing function sends a service API get request to the CAPIF core function, with service API published information reference provided by the CAPIF core function when the service API was published.
2. Upon receiving the service API get request, the CAPIF core function checks whether the API publishing function is authorized to get published service APIs information. If the check is successful, the corresponding service API information is retrieved from the CAPIF core function (API registry).
3. The CAPIF core function provides a service API get response to the API publishing function which includes the service API information.

8.6 Update service APIs

8.6.1 General

The CAPIF core function allows the service API provider to update the information related to the published service API, e.g. a change in the characteristics of the service API. This procedure is initiated by the API publishing function to the CAPIF core function. The API publishing function can be within PLMN trust domain or within 3rd party trust domain.

8.6.2 Information flows

8.6.2.1 Service API update request

Table 8.6.2.1-1 describes the information flow service API update request from the API publishing function to the CAPIF core function.

Table 8.6.2.1-1: Service API update request

Information element	Status	Description
API publisher information	M	The information of the API publisher may include identity, authentication and authorization information
Service API published information reference	M	The information (set) provided by the CAPIF core function about the published service API which can be used for reference by the API publishing function.
Service API information	M	The service API information as specified in Table 8.3.2.1-1 which is required to replace the existing service API information
Reason	O	The reason of the update (e.g. change log)

NOTE: How to monitor service API status when the APF is unable to update service API status is not specified in this release.

8.6.2.2 Service API update response

Table 8.6.2.2-1 describes the information flow service API update response from the CAPIF core function to the API publishing function.

Table 8.6.2.2-1: Service API update response

Information element	Status	Description
Result	M	Indicates the success or failure of updating the service API information
Service API information	O	The authorized service API information during update, applicable when the update result is success.

8.6.3 Procedure

Figure 8.6.3-1 illustrates the procedure for updating the published service APIs information. The service API update mechanism is supported by the CAPIF core function.

Pre-conditions:

1. Authorization details of the APF are available with the CAPIF core function.
2. API invokers may have subscribed with the CAPIF core function to obtain notification regarding update to service API information.

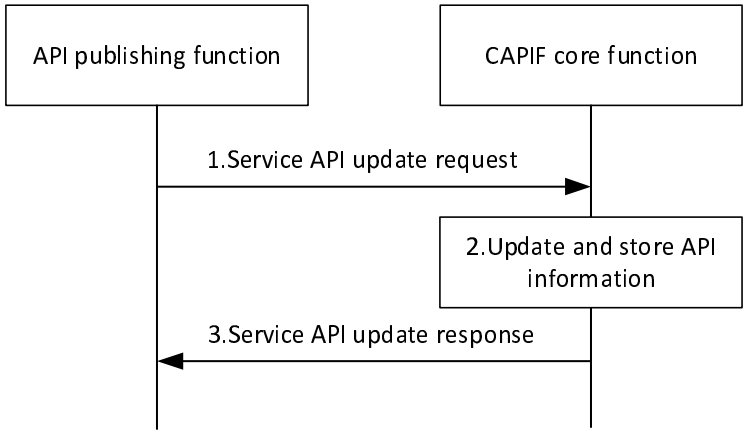

Figure 8.6.3-1: Update service APIs

1. The API publishing function sends a service API update request to the CAPIF core function, which includes the service API published information reference provided by the CAPIF core function when the service API was published and the new service API information which is to be updated.
2. Upon receiving the service API update request, the CAPIF core function checks whether the API publishing function is authorized to update the published service APIs information. If the check is successful, the service API information provided by the API publishing function is updated at the CAPIF core function (API registry).
3. The CAPIF core function provides a service API update response to the API publishing function and triggers notifications to subscribed API invokers as described in subclause 8.8.4.

8.7 Discover service APIs

8.7.1 General

The following procedure in this subclause corresponds to the architectural requirements on discover service APIs.

8.7.2 Information flows

8.7.2.1 Service API discover request

Table 8.7.2.1-1 describes the information flow service API discover request from the API invoker to the CAPIF core function.

Table 8.7.2.1-1: Service API discover request

Information element	Status	Description
API invoker identity information	M	Identity information of the API invoker discovering service APIs
Query information	M	Criteria for discovering matching service APIs (e.g. service API category, Serving Area Information (optional), preferred AEF location (optional), required API provider name (optional), UE IP address (optional), interfaces, protocols), Service KPIs (optional) (see NOTE)
NOTE: It should be possible to discover all the service APIs.		

8.7.2.2 Service API discover response

Table 8.7.2.2-1 describes the information flow service API discover response from the CAPIF core function to the API invoker.

Table 8.7.2.2-1: Service API discover response

Information element	Status	Description
Result	M	Indicates the success or failure of the discovery of the service API information
Service API information (see NOTE 2)	O (see NOTE 1)	List of service APIs corresponding to the request, including service API information such as service API name, API provider name (optional), service API category, communication type, description, Serving Area Information (optional), interface details (e.g. IP address, port number, URI), protocols, version, data format, Service KPIs (optional).
CAPIF core function identity information	O (see NOTE 1)	Indicates the CAPIF core function serving the service API category provided in the query criteria
NOTE 1: The service API information or the CAPIF core function identity information or both shall be present if the Result information element indicates that the service API discover operation is successful. Otherwise both shall not be present. NOTE 2: If topology hiding is enabled for the service API, the interface details shall be the interface details of AEF acting as service communication entry point for the service API.		

8.7.3 Procedure

Figure 8.7.3-1 illustrates the procedure for discover service APIs.

The service API discovery mechanism is supported by the CAPIF core function.

Pre-conditions:

1. The API invoker is onboarded and has received an API invoker identity.
2. The CAPIF core function is configured with a discovery policy information (e.g. to restrict discovery to category of APIs) for API invoker(s).

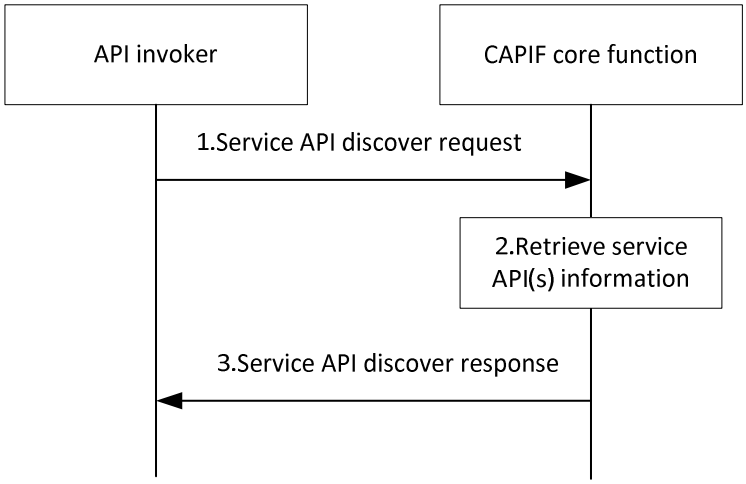

Figure 8.7.3-1: Discover service APIs

1. The API invoker sends a service API discover request to the CAPIF core function. It includes the API invoker identity and query information.

2. Upon receiving the service API discover request, the CAPIF core function verifies the identity of the API invoker (via authentication). The CAPIF core function retrieves the stored service API(s) information from the CAPIF core function (API registry) as per the query information in the service API discover request. Further, the CAPIF core function applies the discovery policy and performs filtering of service APIs information retrieved from the CAPIF core function.

3. The CAPIF core function sends a service API discover response to the API invoker with the list of service API information for which the API invoker has the required authorization.

8.8 Subscription, unsubscription and notifications for the CAPIF events

8.8.1 General

The CAPIF core function enables the subscribing entity (i.e. the API invoker, the API exposing function, the API publishing function, the API management function) to subscribe to and unsubscribe from the CAPIF events such as availability events of service APIs, change in service API information, monitoring service API invocations, API invoker onboarding events, etc. The subscription, unsubscription and notification for the CAPIF events are enabled on the following CAPIF reference points:

- CAPIF-1 or CAPIF-1e: the API invoker can subscribe to and unsubscribe from CAPIF events and receive notifications from the CAPIF core function;

- CAPIF-3 or CAPIF-3e: the AEF can subscribe to and unsubscribe from CAPIF events and receive notifications from the CAPIF core function;

- CAPIF-4 or CAPIF-4e: the API publishing function can subscribe to and unsubscribe from CAPIF events and receive notifications from the CAPIF core function; and

- CAPIF-5 or CAPIF-5e: the API management function can subscribe to and unsubscribe from CAPIF events and receive notifications from the CAPIF core function.

NOTE: Support for subscriptions and notifications can also be part of the actual service APIs. That type of subscriptions and notifications is not covered by the provisions in this clause.

8.8.2 Information flows

8.8.2.1 Event subscription request

Table 8.8.2.1-1 describes the information flow for event subscription request from the subscribing entity to the CAPIF core function.

Table 8.8.2.1-1: Event subscription request

Information element	Status	Description
Identity information	M	The information to determine the identity of the subscribing entity
Event criteria	M	The event criteria include event type information like failure API invocation event, new API available event, API version change event, API location change event, etc and other query information like service API identifier, service API name, etc.
Notification reception information	O	The information of the subscribing entity for receiving the notifications for the event.

8.8.2.2 Event subscription response

Table 8.8.2.2-1 describes the information flow for event subscription response from the CAPIF core function to the subscribing entity.

Table 8.8.2.2-1: Event subscription response

Information element	Status	Description
Result	M	Indicates the success or failure of the event subscription operation
Subscription identifier	O (see NOTE)	The unique identifier for the event subscription.
NOTE: Shall be present if the Result information element indicates that the event subscription operation is successful. Otherwise subscription identifier shall not be present.		

8.8.2.3 Event notification

Table 8.8.2.3-1 describes the information flow for event notification from the CAPIF core function to the subscribing entity. A notification about an event is sent to a subscribing entity if the event criteria in the related subscription match the corresponding attributes of the event content.

Table 8.8.2.3-1: Event notification

Information element	Status	Description
Subscription identifier	M	The unique identifier of the event subscription
Event identifier	M	The unique identifier for the event. For the list of events, refer subclause 8.8.6
Event related information	M	The event related information (e.g. time at which the event originated, location of event)
Event content	M	The content of the event information.

8.8.2.4 Event notification acknowledgement

Table 8.8.2.4-1 describes the information flow event notification acknowledgement from the subscribing entity to the CAPIF core function.

Table 8.8.2.4-1: Event notification acknowledgement

Information element	Status	Description
Acknowledgement	M	Acknowledgement for the event notification received.

8.8.2.5 Event unsubscription request

Table 8.8.2.5-1 describes the information flow for event unsubscription request from the subscribing entity to the CAPIF core function.

Table 8.8.2.5-1: Event unsubscription request

Information element	Status	Description
Identity information	M	The information to determine the identity of the subscribing entity
Subscription identifier	M	The unique identifier for the event subscription that was provided to the subscribing entity during the CAPIF event subscription operation.

8.8.2.6 Event unsubscription response

Table 8.8.2.6-1 describes the information flow for event unsubscription response from the CAPIF core function to the subscribing entity.

Table 8.8.2.6-1: Event unsubscription response

Information element	Status	Description
Result	M	Indicates the success or failure of the event unsubscription operation

8.8.2.7 Event subscription update request

Table 8.8.2.7-1 describes the information flow for event subscription update request from the subscribing entity to the CAPIF core function.

Table 8.8.2.7-1: Event subscription update request

Information element	Status	Description
Identity information	M	The information to determine the identity of the subscribing entity
Subscription identifier	M	The unique identifier for the event subscription that was provided to the subscribing entity during the CAPIF event subscription operation.
Event criteria changes	O (NOTE)	Updates to the event criteria which are defined in clause 8.8.2.1
Notification reception information changes	O (NOTE)	Updates to the information of the subscribing entity for receiving the notifications for the event
NOTE: At least one of these information elements shall be present.		

8.8.2.8 Event subscription update response

Table 8.8.2.8-1 describes the information flow for event subscription update response from the CAPIF core function to the subscribing entity.

Table 8.8.2.2-1: Event subscription update response

Information element	Status	Description
Result	M	Indicates the success or failure of the event subscription update operation

8.8.3 Procedure for CAPIF event subscription

Figure 8.8.3-1 illustrates the procedure for CAPIF events subscription.

Pre-conditions:

1. The subscribing entity has the authorization to subscribe for the CAPIF events.

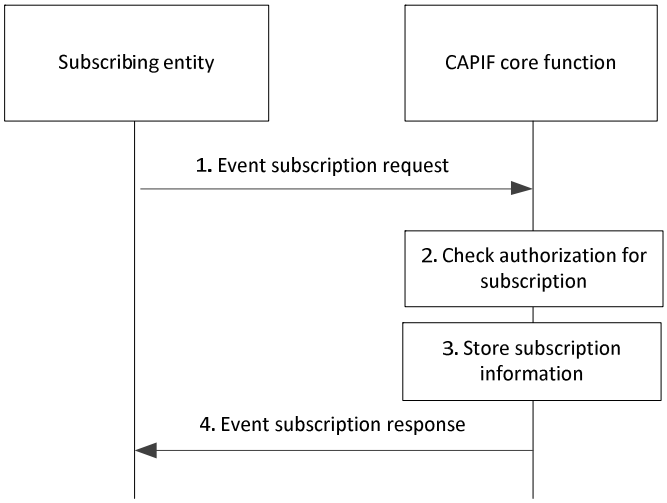

Figure 8.8.3-1: Procedure for CAPIF event subscription

1. The subscribing entity sends an event subscription request to the CAPIF core function in order to receive notification of events.
2. Upon receiving the event subscription request from the subscribing entity, the CAPIF core function checks for the relevant authorization for the event subscription.
3. If the authorization is successful, the CAPIF core function stores the subscription information.
4. The CAPIF core function sends an event subscription response indicating successful operation.

8.8.4 Procedure for CAPIF event notifications

Figure 8.8.4-1 illustrates the procedure for CAPIF event notifications.

Pre-conditions:

1. The subscription procedure as illustrated in figure 8.8.3-1 is performed by the subscribing entity.

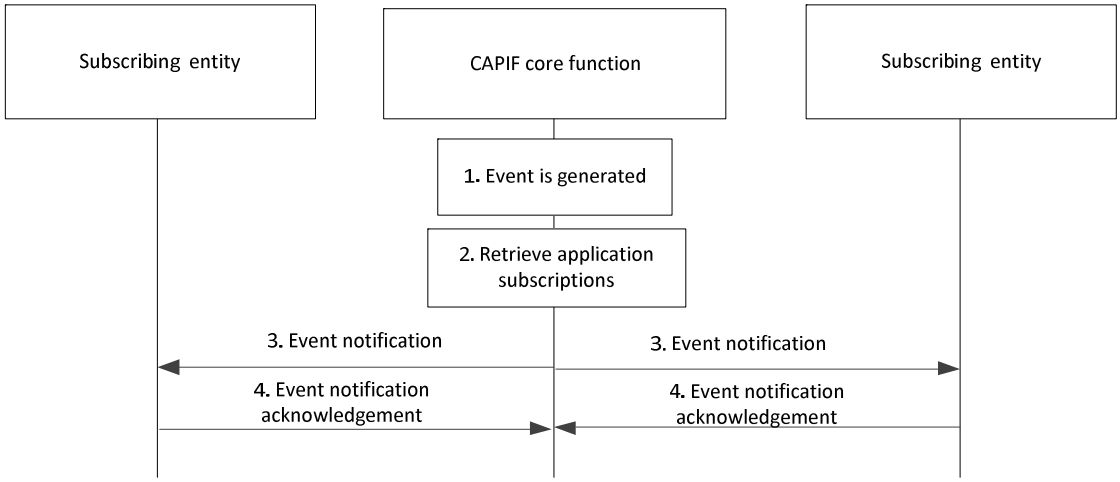

Figure 8.8.4-1: Procedure for CAPIF event notifications

1. The CAPIF core function generates events to be consumed by the subscribing entity(s).
2. For the generated event, the CAPIF core function retrieves the list of corresponding subscriptions.
3. The CAPIF core function sends event notifications to all the subscribing entity(s) that have subscribed for the event matching the criteria. If a notification reception information is available as part of the subscribing entity event subscription, then the notification reception information is used by the CAPIF core function to send event notifications to the subscribing entity.
4. The subscribing entity sends an event notification acknowledgement to the CAPIF core function for the event notification received.

8.8.5 Procedure for CAPIF event unsubscription

Figure 8.8.5-1 illustrates the procedure for CAPIF event unsubscription.

Pre-condition:

1. The subscribing entity has subscribed to the CAPIF events.

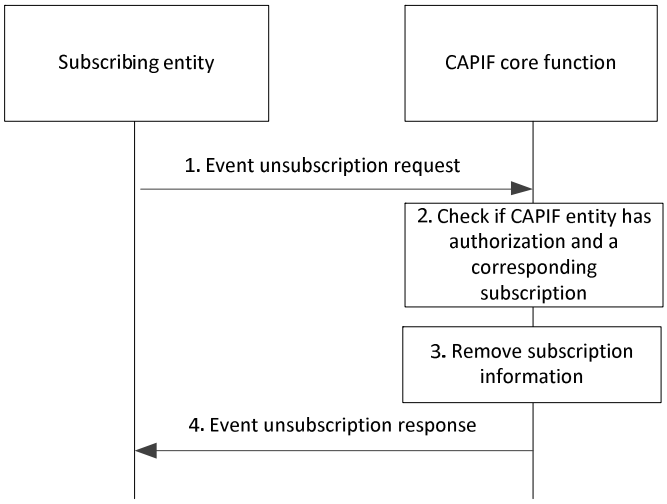

Figure 8.8.5-1: Procedure for CAPIF event unsubscription

1. The subscribing entity sends an event unsubscription request to the CAPIF core function with the information of the subscribed CAPIF event.

2. Upon receiving the event unsubscription request from the subscribing entity, the CAPIF core function checks for the event subscription corresponding to the subscribing entity and further checks if the subscribing entity is authorized to unsubscribe from the CAPIF event.
3. If the event subscription information corresponding to the subscribing entity is available and the subscribing entity is authorized to unsubscribe for the CAPIF event, the CAPIF core function removes the subscription information.
4. The CAPIF core function sends an event unsubscription response indicating successful operation.

8.8.5a Procedure for CAPIF event subscription update

Figure 8.8.5a-1 illustrates the procedure for CAPIF events subscription update.

Pre-conditions:

1. The subscribing entity has the authorization to update subscriptions for CAPIF events.
2. The subscribing entity has created subscriptions for CAPIF events.

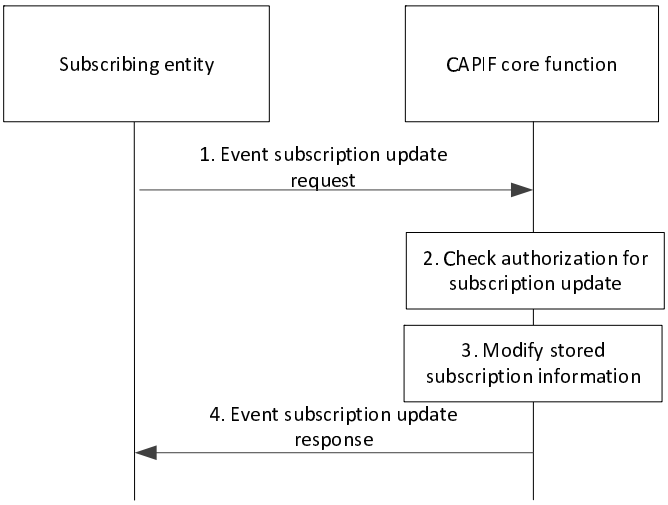

Figure 8.8.5a-1: Procedure for CAPIF event subscription

1. The subscribing entity sends an event subscription update request to the CAPIF core function in order update a previous subscription to receive notification of events.
2. Upon receiving the event subscription update request from the subscribing entity, the CAPIF core function checks for the relevant authorization for the event subscription update.
3. If the authorization is successful, the CAPIF core function updates the subscription information.
4. The CAPIF core function sends an event subscription update response indicating successful operation.

8.8.6 List of CAPIF events

Table 8.8.6-1 provides a non-exhaustive list of CAPIF events.

Table 8.8.6-1: List of CAPIF events

Events	Events Description
Availability of service APIs	Availability events of service APIs (e.g. active, inactive)
Service API updated	Events related to change in service API information
Monitoring service API invocations	Events corresponding to service API invocations
API invoker status	Events related to API invoker status in CAPIF (onboarded, offboarded)
API topology hiding status	Events related to API topology hiding status in CAPIF (created, revoked)
System related events	Alarm events providing fault information
Performance related events	Events related to system load conditions

8.9 Revoking subscription of the CAPIF events

8.9.1 General

The CAPIF core function allows to revoke subscription of CAPIF events for the subscribing entity related to the service API changes, such as availability events of service APIs, change in service API information, monitoring service API invocations, API invoker onboarding events, etc. This procedure is initiated by the CAPIF core function.

NOTE: It is optional to trigger notification by the CAPIF core function for revocation of subscription for CAPIF event(s).

8.9.2 Information flows

This subclause describes the information flows for CAPIF event subscription revocation.

8.9.2.1 Subscription revoke notification

Table 8.9.2.1-1 describes the information flow for subscription revoke notification from the CAPIF core function to the subscribing entity.

Table 8.9.2.1-1: Subscription revoke notification

Information element	Status	Description
Identity information	M	The information to determine the identity of the subscribing entity
Subscription identifier	M	The unique identifier for the event subscription that was provided to the subscribing entity during the CAPIF event subscription operation.
Reason	O	Indicate the reason of subscription revocation

8.9.2.2 Subscription revoke notification acknowledgement

Table 8.9.2.2-1 describes the information flow for subscription revoke notification acknowledgement from the subscribing entity to the CAPIF core function.

Table 8.9.2.2-1: Subscription revoke notification acknowledgement

Information element	Status	Description
Acknowledgement	M	The acknowledgement for the received notification.

8.9.3 Procedure

Figure 8.9.3-1 illustrates the procedure for subscription revocation, triggered by the CAPIF core function.

Pre-conditions:

1. The subscribing entity has previously subscribed to CAPIF event(s) to the CAPIF core function.

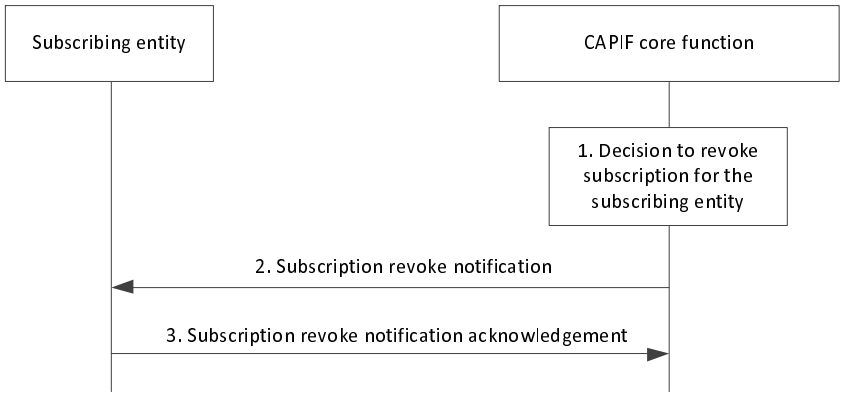

Figure 8.9.3-1: Procedure for revoking subscription of the CAPIF events

1. The CAPIF core function decides to revoke subscription of CAPIF event(s) for the subscribing entity.
2. The CAPIF core function sends subscription revoke notification to the subscribing entity.
3. The subscribing entity provides a subscription revoke notification acknowledgement to the CAPIF core function.

8.10 Authentication between the API invoker and the CAPIF core function

8.10.1 General

The procedure in this subclause corresponds to the architectural requirements for authentication between the API invoker and the CAPIF core function.

8.10.2 Information flows

NOTE: The security aspects of this procedure are specified in subclause 6.2 and subclause 6.3.1 of 3GPP TS 33.122 [12].

8.10.3 Procedure

Figure 8.10.3-1 illustrates the procedure for authentication between the API invoker and the CAPIF core function.

Pre-conditions:

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

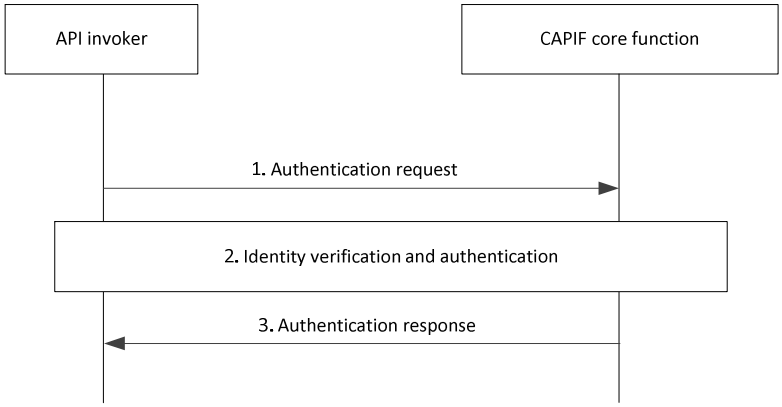

Figure 8.10.3-1: Procedure for authentication between the API invoker and the CAPIF core function

1. The API invoker triggers authentication to the CAPIF core function, including the identity confirmed after successful onboarding.

2. Upon receiving the authentication request, the CAPIF core function verifies the identity with the API invoker profile and authenticates the API invoker.

NOTE 1: The authentication process is specified in subclause 6.2 and subclause 6.3.1 of 3GPP TS 33.122 [12].

3. The CAPIF core function returns the result of the API invoker identity verification in the authentication response.

NOTE 2: The CAPIF core function can share the information required for authentication of the API invoker at the AEF.

8.11 API invoker obtaining authorization to access service API

8.11.1 General

The API invoker requires to execute this procedure when it needs to obtain or re-obtain (e.g. upon expiry of the authorization information) the authorization to access the service API. Once the API invoker receives the authorization to access the service API, the API invoker can perform one or multiple service API invocations as per the permission limit. This procedure may be performed during the API invoker onboarding process.

8.11.2 Information flows

NOTE: The security aspects of this procedure are specified in subclause 6.5.2.3 of 3GPP TS 33.122 [12].

8.11.3 Procedure

Figure 8.11.3-1 illustrates the procedure for obtaining authorization to access the service API.

Pre-condition:

1. The API invoker is onboarded and has received an API invoker identity.

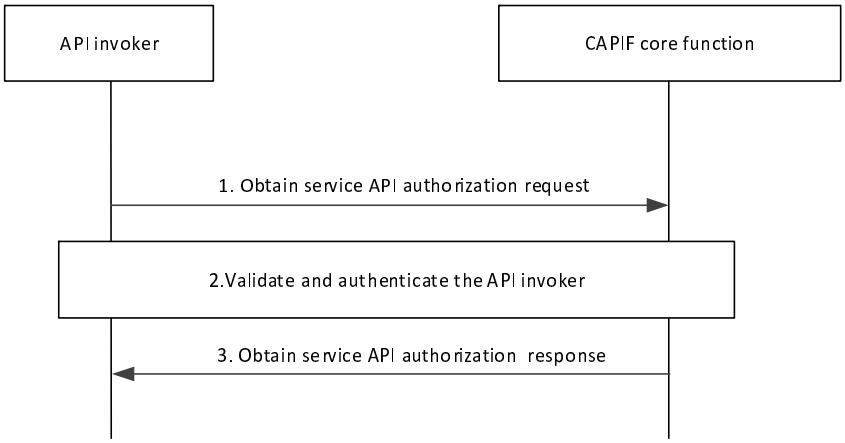

Figure 8.11.3-1: Procedure for the API invoker obtaining authorization for service API access

1. The API invoker sends an obtain service API authorization request to the CAPIF core function for obtaining permission to access the service API by including the API invoker identity information and any information required for authentication of the API invoker.

2. The CAPIF core function validates the authentication of the API invoker (using authentication information) and checks whether the API invoker is permitted to access the requested service API.

NOTE 1: The authentication process is specified in subclause 6.5.2.3 of 3GPP TS 33.122 [12].

3. Based on the API invoker's subscription information the authorization information to access the service APIs is sent to the API invoker in the obtain service API authorization response.

NOTE 2: The mechanism for distribution of the authorization information for the API invoker to the API exposing function is specified in subclause 6.5.2.3 of 3GPP TS 33.122 [12].

8.12 AEF obtaining service API access control policy

8.12.1 General

The CAPIF core function is the central repository of all the policies related to service APIs. The AEF executes this procedure when it needs to obtain the policy to perform access control on the service API invocations (e.g. when policy for performing access control on service API is unavailable at the AEF). The AEF can be within PLMN trust domain or within 3rd party trust domain.

8.12.2 Information flows

8.12.2.1 Obtain access control policy request

Table 8.12.2.1-1 describes the information flow obtain access control policy request from the AEF to the CAPIF core function.

Table 8.12.2.1-1: Obtain access control policy request

Information element	Status	Description
Identity information	M	Identity information of the entity requesting the access control policy
Service API identification	M	The identification information of the service API for which the access control policy is being requested.

8.12.2.2 Obtain access control policy response

Table 8.12.2.2-1 describes the information flow obtain access control policy response from the CAPIF core function to the AEF.

Table 8.12.2.2-1: Obtain access control policy response

Information element	Status	Description
Result	M	Indicates the success or failure of the obtain access control policy operation
Access control policy information	O (see NOTE)	The access control policy information corresponding to the requested service API.
NOTE: Shall be present if the Result information element indicates that the obtain access control policy operation is successful. Otherwise access control policy information shall not be present.		

8.12.3 Procedure

Figure 8.12.3-1 illustrates the procedure for obtaining policy to perform access control on the service API invocations.

Pre-conditions:

1. The AEF is hosting the service API but the policy to perform access control is not available with AEF.
2. The CAPIF core function is configured with the access control policies corresponding to one or more service APIs.
3. Authorization details of the AEF are available with the CAPIF core function.

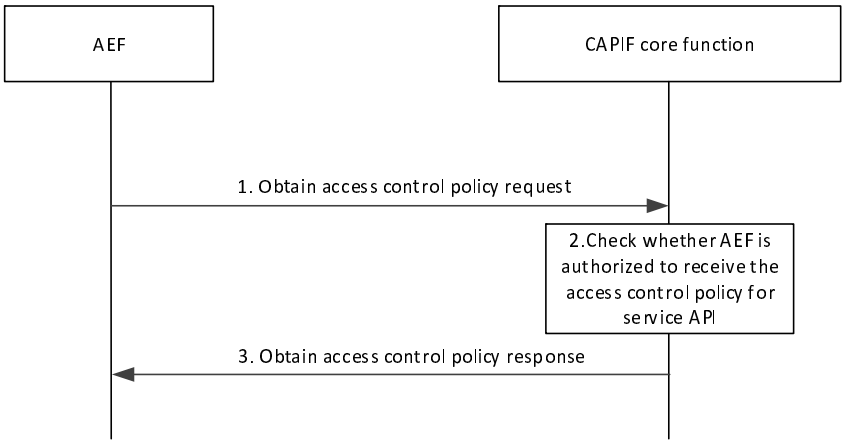

Figure 8.12.3-1: Procedure for the AEF obtaining service API access control policy

1. The AEF sends an obtain access control policy request to the CAPIF core function for obtaining the policy to perform the access control on service API invocations by including the details of the hosted service API.
2. The CAPIF core function checks whether the AEF is authorized to receive the access control policy corresponding to the service APIs requested.
3. If authorization check is successful, the AEF is provided the access control policy for the service API via an obtain access control policy response. If authorization check is not successful, the AEF is provided with a failure indication via a obtain access control policy response.

NOTE: To maintain synchronization between the AEF and the CAPIF core function for the policy cached at AEF, the AEF can subscribe to the policy update event at CAPIF core function according to the procedure in subclause 8.8.3 and receive notifications about any updated policy at CAPIF core function according to the procedure in subclause 8.8.4.

8.13 Topology hiding

8.13.1 General

The procedure in this subclause corresponds to the architectural requirements for hiding the topology of the PLMN trust domain from the API invokers accessing the service APIs from outside the PLMN trust domain.

8.13.2 Information flows

8.13.2.1 Service API invocation request (API invoker – AEF-1)

The information flow service API invocation request from the API invoker to AEF-1 (AEF acting as service communication entry point) is service API specific and the complete detail of the service API invocation request is out of scope of the present document. Table 8.17.2.1-1 describes the CAPIF related information elements which are included in the service API invocation request.

8.13.2.2 Service API invocation request (AEF-1 – AEF-2)

The information flow service API invocation request from AEF-1 (AEF acting as service communication entry point) to AEF-2 (destination AEF for handling service API) is service API specific and the complete detail of the service API invocation request is out of scope of the present document. Table 8.17.2.1-1 describes the CAPIF related information elements which are included in the service API invocation request.

8.13.2.3 Service API invocation response (AEF-2 – AEF-1)

The information flow service API invocation response from AEF-2 (destination AEF for handling service API) to AEF-1 (AEF acting as service communication entry point) is service API specific and the complete detail of the service API invocation response is out of scope of the present document. Table 8.17.2.2-1 describes the CAPIF related information elements which are included in the service API invocation response.

8.13.2.4 Service API invocation response (AEF-1 – API invoker)

The information flow service API invocation response from AEF-1 (AEF acting as service communication entry point) to the API invoker is service API specific and the complete detail of the service API invocation response is out of scope of the present document. Table 8.17.2.2-1 describes the CAPIF related information elements which are included in the service API invocation response.

8.13.3 Procedure

Figure 8.13.3-1 illustrates the procedure for CAPIF topology hiding.

Pre-conditions:

1. The API invoker has performed the service discovery and received the details of the service API which includes the information about the service communication entry point of the AEF-1 in the CAPIF.
2. The API invoker is authenticated and authorized to use the service API.
3. The AEF-1 in the CAPIF is configured with a policy for topology hiding including the entry point address of the service API (provided via AEF-2).

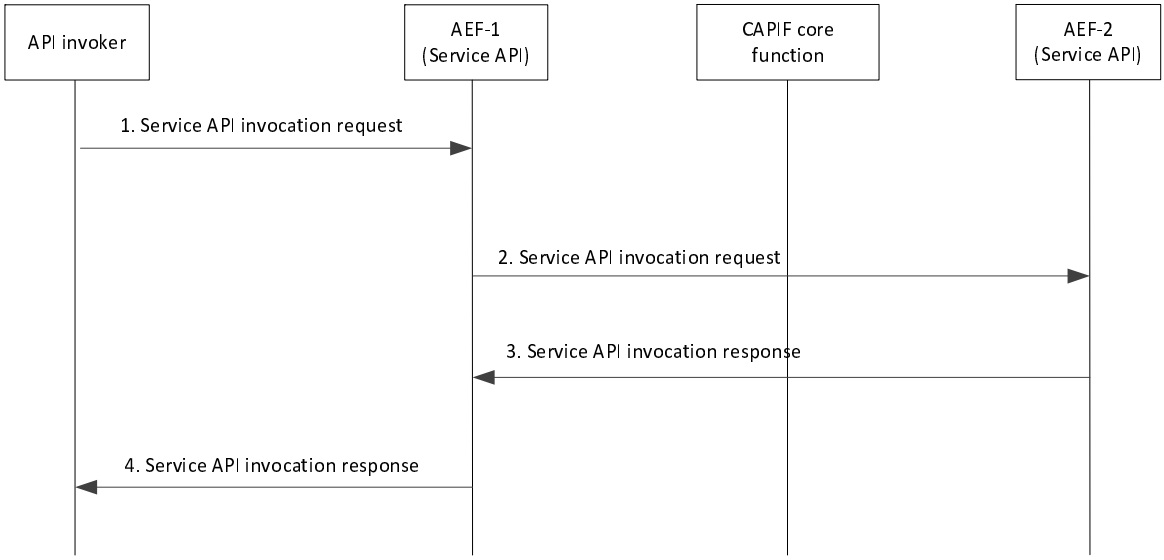

Figure 8.13.3-1: Procedure for CAPIF topology hiding

1. The API invoker performs service API invocation according to the interface of the service API by sending a service API invocation request towards the AEF-1 which exposes the service API towards the API invoker, and acts as topology hiding entity.

NOTE: Steps 2 and 3 are not necessary when the AEF-1 is capable to serve the service API invocation request.

2. The AEF-1 further resolves the actual destination service API address information according to the topology hiding policy and forwards the incoming service API invocation request to the service API of the related AEF-2.

3. The AEF-1 receives a response request for service API invocation from service API provided by AEF-2.

4. The AEF-1 resolves the destination API invoker address and also modifies the source address information of the AEF-2 within the response request as per topology hiding policy and forwards the response request to the API invoker.

8.14 Authentication between the API invoker and the AEF prior to service API invocation

8.14.1 General

The procedure in this subclause corresponds to the architectural requirements for authentication of the API invoker by the AEF.

To reduce latency during API invocation, the API invoker associated authentication information can be made available at the AEF after authentication between the API invoker and the CAPIF core function.

8.14.2 Information flows

NOTE: The security aspects of this procedure are specified in subclause 6.4 and subclause 6.5.2 of 3GPP TS 33.122 [12].

8.14.3 Procedure

Figure 8.14.3-1 illustrates the procedure for authentication between the API invoker and the AEF.

Pre-conditions:

1. Optionally, the CAPIF core function has shared the information required for authentication of the API invoker with the AEF.

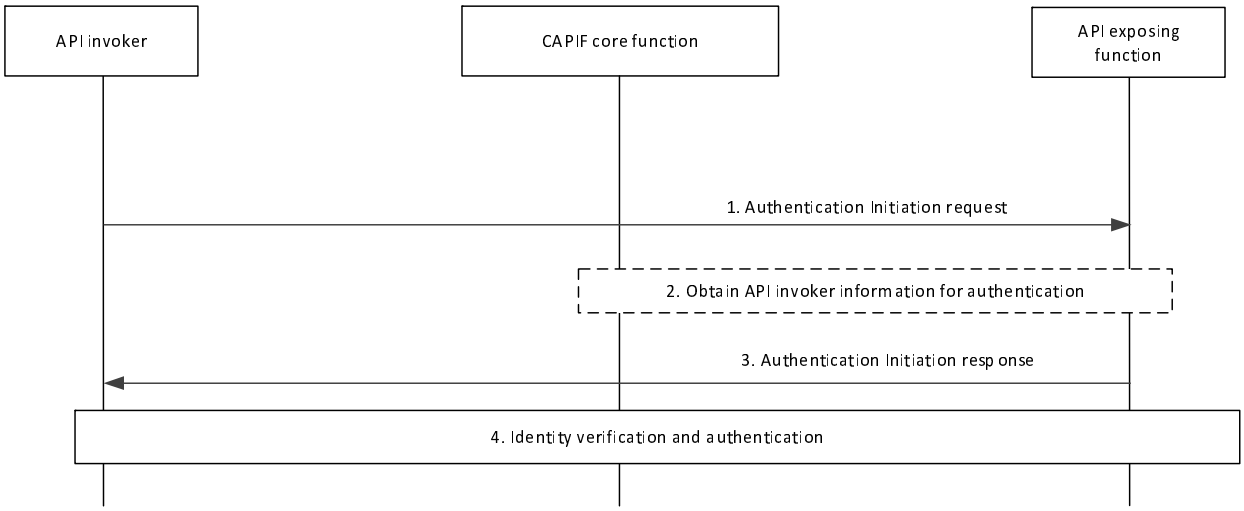

Figure 8.14.3-1: Procedure for authentication between the API invoker and the AEF prior to service API invocation

1. The API invoker triggers authentication initiation to the AEF, including the API invoker identity.
2. The AEF obtains the API invoker information required for authentication by the AEF, if not available.
3. The AEF returns the result of authentication initiation in the authentication initiation response.
4. The AEF verifies the identity of the API invoker and authenticates the API invoker.

NOTE 1: The authentication process is specified in subclause 6.4 and subclause 6.5.2 of 3GPP TS 33.122 [12].

NOTE 2: The authentication is terminated at the AEF acting as the service communication entry point when topology hiding is enabled for the service API.

8.15 Authentication between the API invoker and the AEF upon the service API invocation

8.15.1 General

The procedure in this subclause corresponds to the architectural requirements for authentication of the API invoker by the AEF upon the service API invocation.

To reduce latency during API invocation, the API invoker associated authentication information can be made available at the AEF after authentication between the API invoker and the CAPIF core function.

8.15.2 Information flows

NOTE: The security aspects of this procedure are specified in subclause 6.5.2.3 of 3GPP TS 33.122 [12].

8.15.2.1 Service API invocation request with authentication information

The information flow service API invocation request with authentication information from the API invoker to the AEF is service API specific and the complete detail of the service API invocation request is out of scope of the present document. Table 8.15.2.1-1 describes only the CAPIF related information elements which are included in the service API invocation request.

Table 8.15.2.1-1: Service API invocation request with authentication information

Information element	Status	Description
API invoker identity information	M	The information that determines the identity of the API invoker
Authentication information	M (see NOTE)	The authentication information obtained before initiating the service API invocation request
Service API identification	M	The identification information of the service API for which invocation is requested. The service API identification is part of the specific service API invocation request.
NOTE: The specific aspect of this information element is specified in subclause 6.5.2.3 of 3GPP TS 33.122 [12].		

8.15.2.2 Service API invocation response

The information flow service API invocation response from the AEF to the API invoker is service API specific and the complete detail of the service API invocation response is out of scope of the present document. Table 8.15.2.2-1 describes only the CAPIF related information elements which are included in the service API invocation response.

Table 8.15.2.2-1: Service API invocation response

Information element	Status	Description
Result	M	Indicates the success or failure of service API invocation.

8.15.3 Procedure

Figure 8.15.3-1 illustrates the procedure for authentication of the API invoker by the AEF, where the authentication information is carried in the API invocation request.

Pre-conditions:

1. Optionally, the CAPIF core function has shared the information required for authentication of the API invoker with the AEF.

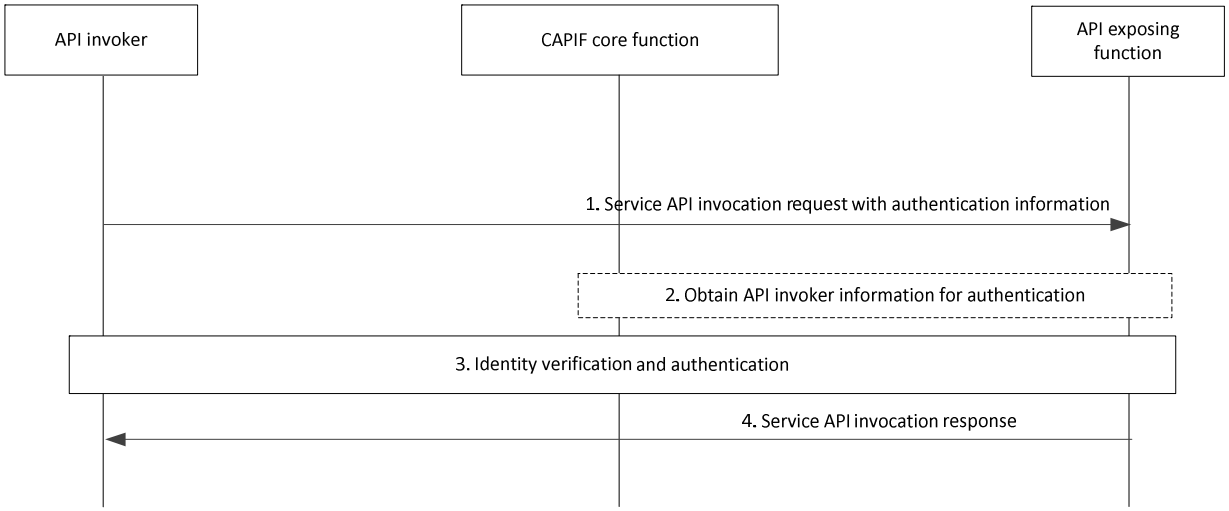

Figure 8.15.3-1: Procedure for authentication between the API invoker and the AEF upon the service API invocation

1. The API invoker invokes a service API invocation request with authentication information to the AEF, and includes in this request authentication information, including the API invoker identity.
2. The AEF obtains the API invoker information required for authentication by the AEF, if not available.
3. The AEF verifies the identity of the API invoker and authenticates the API invoker.

NOTE 1: The authentication process is specified in subclause 6.5.2.3 of 3GPP TS 33.122 [12].

4. If the verification was successful, the AEF returns the result of the service API invocation in the Service API invocation response.

NOTE 2: The authentication is terminated at the AEF acting as the service communication entry point when topology hiding is enabled for the service API.

8.16 Service API invocation with AEF authorization

8.16.1 General

The procedure in this subclause corresponds to the architectural requirements to validate authorization of API invokers upon the service API invocation.

To reduce latency during API invocation, the API invoker associated authorization information can be made available at the AEF after authentication between the API invoker and the CAPIF core function.

NOTE: The security aspects of service API invocation are specified in TS 33.122 [12] clause 6.4 (CAPIF-2) and 6.5 (CAPIF-2e).

8.16.2 Information flows

8.16.2.1 Service API invocation request

The information flow service API invocation request from the API invoker to the AEF is service API specific and the complete detail of the service API invocation request is out of scope of the present document. Table 8.16.2.1-1 describes only the CAPIF related information elements which are included in the service API invocation request.

Table 8.16.2.1-1: Service API invocation request

Information element	Status	Description
API invoker identity information	M	The information that determines the identity of the API invoker
Authorization information	O (see NOTE)	The authorization information obtained before initiating the service API invocation request
Service API identification	M	The identification information of the service API for which invocation is requested. The service API identification is part of the specific service API invocation request.
NOTE: The inclusion of this information element depends on the chosen solution for authorization.		

8.16.2.2 Service API invocation response

The information flow service API invocation response from the AEF to the API invoker is service API specific and the complete detail of the service API invocation response is out of scope of the present document. Table 8.16.2.2-1 describes only the CAPIF related information elements which are included in the service API invocation response.

Table 8.16.2.2-1: Service API invocation response

Information element	Status	Description
Result	M	Indicates the success or failure of service API invocation.

8.16.3 Procedure

Figure 8.16.3-1 illustrates the procedure for API invoker authorization to access service APIs.

Pre-conditions:

1. The API invoker has been authenticated.
2. The API invoker associated authorization information is available at AEF.

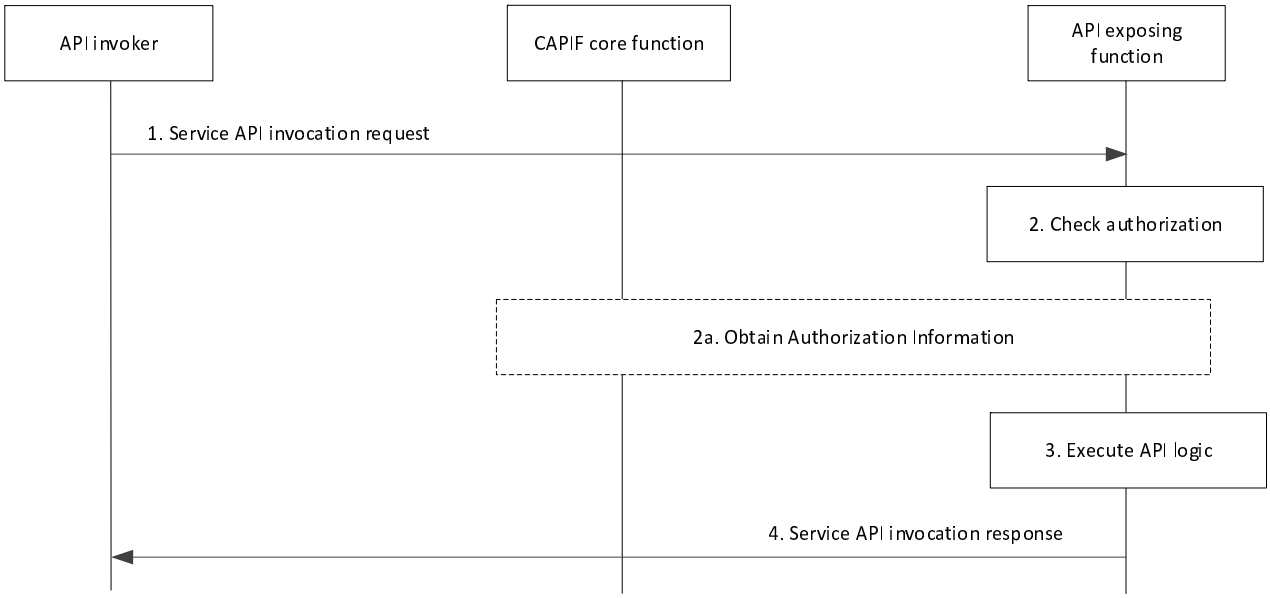

Figure 8.16.3-1: Procedure for API invoker authorization to access service APIs

1. The API invoker triggers service API invocation request to the AEF, including the service API to be invoked.

NOTE 1: Authentication can also be performed if not authenticated previously.

NOTE 2: The API invoker can trigger several service API invocations asynchronously.

2. Upon receiving the service API invocation request, the AEF checks whether the API invoker is authorized to invoke that service API, based on the authorization information.

2a. If the AEF does not have information required to authorize service API invocation, the AEF obtains the authorization information from the CAPIF core function.

3. The AEF executes the service logic for the invoked service API.

4. The API invoker receives the service API invocation response as a result of the service API invocation.

8.17 CAPIF access control

8.17.1 General

The CAPIF controls the access of service API by the API invoker based on policy or usage limits.

8.17.2 Information flows

8.17.2.1 Service API invocation request

The information flow service API invocation request from the API invoker to the AEF is service API specific and the complete detail of the service API invocation request is out of scope of the present document. Table 8.17.2.1-1 describes only the CAPIF related information elements which are included in the service API invocation request.

Table 8.17.2.1-1: Service API invocation request

Information element	Status	Description
API invoker identity information	M	The information that determines the identity of the API invoker
Authorization information	O (see NOTE)	The authorization information obtained before initiating the service API invocation request
Service API identification	M	The identification information of the service API for which invocation is requested. The service API identification is part of the specific service API invocation request.
NOTE: The inclusion of this information element depends on the chosen solution for authorization.		

8.17.2.2 Service API invocation response

The information flow service API invocation response from the AEF to the API invoker is service API specific and the complete detail of the service API invocation response is out of scope of the present document. Table 8.17.2.2-1 describes only the CAPIF related information elements which are included in the service API invocation response.

Table 8.17.2.2-1: Service API invocation response

Information element	Status	Description
Result	M	Indicates the success or failure of service API invocation.

8.17.3 Procedure

Figure 8.17.3-1 illustrates the procedure for service API access control.

Pre-conditions:

1. The API invoker has performed the service API discovery and received the details of the service API which includes the information about the service communication entry point of the AEF in the CAPIF.
2. The API invoker is authenticated and authorized to use the service API.
3. The AEF in the CAPIF is configured with at least one access policy to be applied to the service API invocation corresponding to the API invoker and service API.

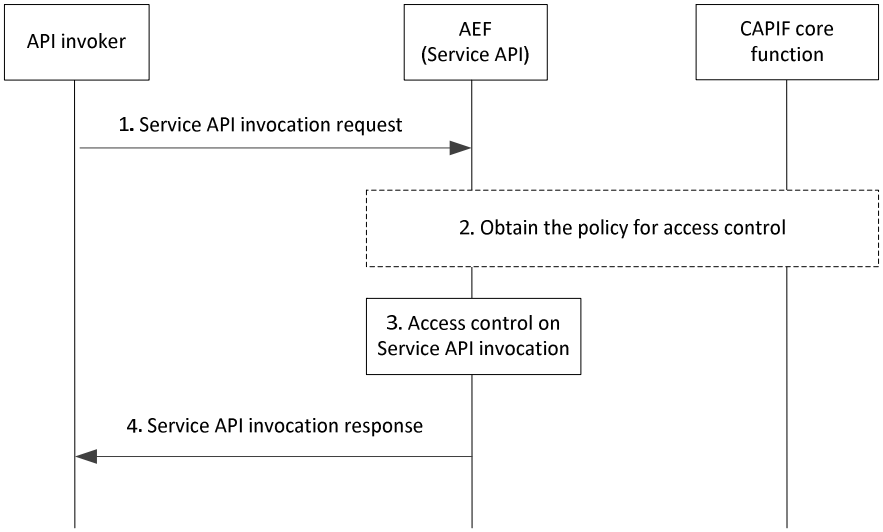

Figure 8.17.3-1: Procedure for service API access control

1. The API invoker performs service API invocation according to the interface of the service API by sending a service API invocation request towards the AEF which exposes the service API towards the API invoker. The AEF acts as an access control entity.
2. If the access control policy is not configured with AEF, then the AEF may obtain the access control policy configuration from the CAPIF core function.
3. Upon receiving the service API invocation request from the API invoker, the AEF checks for configuration for access control. As per the configuration for access control, the AEF performs access control on the service API invocation request as per the operator policy.
4. The API invoker receives a service API invocation response for service API invocation from the AEF providing the service API.

8.18 CAPIF access control with cascaded AEFs

8.18.1 General

The procedure in this subclause corresponds to the architectural requirements related to some common access control requirements for service API invocations. It provides access control, based on two cascaded API Exposing Function (AEF) instances. While one AEF instance provides the entry point for the service API and acts as access controller, further AEF instances deliver the functionality of the actual service APIs.

8.18.2 Information flows

8.18.2.1 Service API invocation request

The information flow service API invocation request from the API invoker to the AEF and between AEFs is service API specific and the complete detail of the service API invocation request is out of scope of the present document. Table 8.17.2.1-1 describes the CAPIF related information elements which are included in the service API invocation request.

8.18.2.2 Service API invocation response

The information flow service API invocation response from the AEF to the API invoker and between AEFs is service API specific and the complete detail of the service API invocation response is out of scope of the present document.

Table 8.17.2.2-1 describes the CAPIF related information elements which are included in the service API invocation response.

8.18.3 Procedure

Figure 8.18.3-1 illustrates the procedure for CAPIF access control.

Pre-conditions:

1. The API invoker has performed the service discovery and received the details of the service API which includes the information about the service communication entry point of the AEF-1 in the CAPIF.
2. The API invoker is authenticated and authorized to use the service API.
3. The AEF-1 in the CAPIF is configured with at least one access policy to be applied to the service API invocation corresponding to the API invoker and service API.

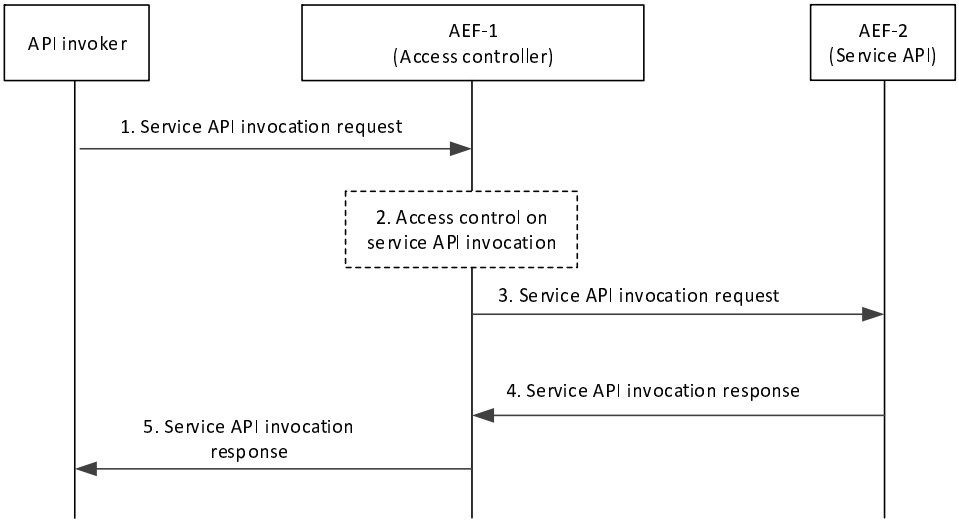

Figure 8.18.3-1: Procedure for CAPIF access control with cascaded AEFs

1. The API invoker performs service API invocation according to the interface of the service API by sending a service API invocation request towards the AEF-1 which exposes the service API towards the API invoker, and acts as access control entity.
2. Upon receiving the service API invocation request from the API invoker, the AEF-1 checks for configuration for access control. As per the configuration for access control, the AEF-1 performs access control on the service API invocation as per the operator policy.
3. The AEF-1 forwards the incoming service API invocation request to the service API provided by AEF-2.
4. The AEF-1 receives a service API invocation response for service API invocation from AEF-2.
5. The AEF-1 resolves the destination API invoker address and modifies the source address information of AEF-2 within the service API invocation response and forwards the service API invocation response to the API invoker.

8.19 Logging service API invocations

8.19.1 General

The procedure in this subclause corresponds to the architectural requirements for logging service API invocations at AEF. The AEF can be within PLMN trust domain or within 3rd party trust domain.

8.19.2 Information flows

8.19.2.1 API invocation log request

Table 8.19.2.1-1 describes the information flow API invocation log request from the API exposing function to the CAPIF core function.

Table 8.19.2.1-1: API invocation log request

Information element	Status	Description
API exposing identity information	M	Identity information of the AEF logging service API(s) invocations
API invocation log information	M	API invocation log information such as API invoker's ID, IP address, service API name, version, invoked operation, input parameters, invocation result, time stamp information

8.19.2.2 API invocation log response

Table 8.19.2.2-1 describes the information flow API invocation log response from the CAPIF core function to the API exposing function.

Table 8.19.2.2-1: API invocation log response

Information element	Status	Description
Result	M	Indicates the success or failure of API(s) invocation log request

8.19.3 Procedure

Figure 8.19.3-1 illustrates the procedure for logging service API invocations at AEF.

Pre-conditions:

1. The API invoker(s) has invoked certain service API(s).
2. Authorization details of the AEF are available with the CAPIF core function.

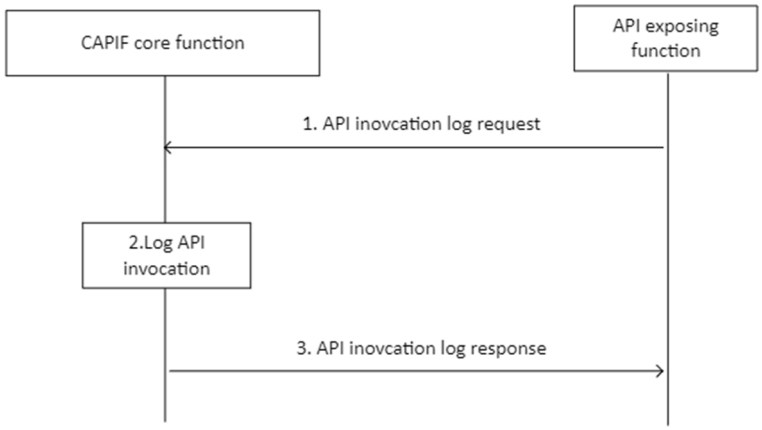

Figure 8.19.3-1: Procedure for logging service API invocations

1. Upon invocation of service API(s) from one more API invokers, the AEF triggers API invocation log request towards the CAPIF core function.

NOTE 1: The AEF can collect the log information associated to several API invocations before triggering API invocation log request asynchronously.

2. The CAPIF core function makes a log entry and stores the information e.g. for charging purposes, for access by authorized users and entities.

NOTE 2: API invocation log is stored for a configured duration.

3. AEF receives the API invocation log response from the CAPIF core function.

8.20 Charging the invocation of service APIs

8.20.1 General

The procedure in this subclause corresponds to the architectural requirements for charging the invocation of service APIs. The AEF can be within PLMN trust domain or within 3rd party trust domain.

8.20.2 Information flows

NOTE: It is in SA5 scope to develop the charging related information flows for this procedure.

Editor's note: Reference to the appropriate SA5 specification is needed.

8.20.3 Procedure

Figure 8.20.3-1 illustrates the procedure for charging the invocation of service APIs.

Pre-conditions:

1. Authorization details of the AEF are available with the CAPIF core function.

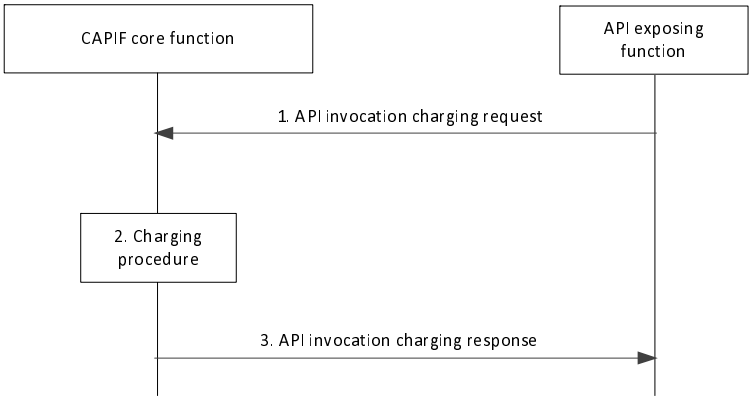

Figure 8.20.3-1: Procedure for charging the invocation of service APIs

1. Upon invocation of service API(s) from one more API invokers, the AEF triggers an API invocation charging request and includes API invoker information (e.g. invoker's ID and IP address, location, timestamp) and service API information (e.g. service API name and version, invoked operation, input parameters, invocation result) towards the CAPIF core function.

NOTE: These requests can be triggered asynchronously.

2. The CAPIF core function performs a charging procedure which includes storing the information for access by authorized API management.

3. The AEF receives the API invocation charging response from the CAPIF core function.

8.21 Monitoring service API invocation

8.21.1 General

The procedure in this subclause corresponds to the architectural requirements for monitoring service API invocation.

8.21.2 Information flows

8.21.2.1 Monitoring service API event notification

The information flow for the monitoring service API event notification from the CAPIF core function to the API management function is same as the event notification from the CAPIF core function to the subscribing entity. Table 8.8.2.3-1 describes the information elements which are included in the monitoring service API event notification.

8.21.2.2 Monitoring service API event notification acknowledgement

The information flow for the monitoring service API event notification acknowledgement from the API management function to the CAPIF core function is same as the event notification acknowledgement from subscribing entity to the CAPIF core function. Table 8.8.2.4-1 describes the information elements which are included in the monitoring service API event notification acknowledgement.

8.21.3 Procedure

Figure 8.21.3-1 illustrates the procedure for monitoring service API invocation.

Pre-conditions:

1. The API management function has subscribed to monitoring event including filters such as invoker's ID and IP address, service API name and version, input parameters, and invocation result.

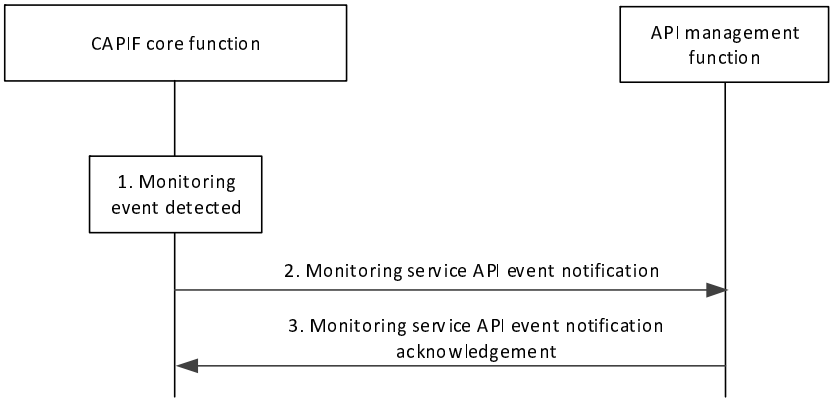

Figure 8.21.3-1: Procedure for monitoring service API invocation

1. The CAPIF core function monitors the service API invocations applying the monitoring filters specified before.
2. Detection of a monitoring event by the CAPIF core function triggers notification to the API management function with the details of the monitored event.

NOTE: API provider action subsequent to monitoring service API notification is out-of-scope of this specification.

3. The API management function sends a monitoring service API event notification acknowledgement to the CAPIF core function for the notification received.

8.22 Auditing service API invocation

8.22.1 General

The procedure in this subclause corresponds to the architectural requirements for auditing service API invocation. This procedure can be used for auditing of other CAPIF interactions i.e. service API invocation events, API invoker onboarding events and API invoker interactions with the CAPIF (e.g. authentication, authorization, discover service APIs) as well. The API management function can be within PLMN trust domain or within 3rd party trust domain.

8.22.2 Information flows

8.22.2.1 Query service API log request

Table 8.22.2.1-1 describes the information flow query service API log request from the API management function to the CAPIF core function.

Table 8.22.2.1-1: Query service API log request

Information element	Status	Description
Identity information	M	Identity information of the entity querying service API log request
Query information	M	List of query filters such as invoker's ID and IP address, service API name and version, input parameters, and invocation result

8.22.2.2 Query service API log response

Table 8.22.2.2-1 describes the information flow query service API log response from the CAPIF core function to the API management function.

Table 8.22.2.2-1: Query service API log response

Information element	Status	Description
Result	M	Indicates the success or failure of query service API log request
API invocation log information	O (see NOTE)	API invocation log information such as API invoker's ID, IP address, service API name, version, invoked operation, input parameters, invocation result, time stamp information
NOTE: Information element shall be present when result indicates success.		

8.22.3 Procedure

Figure 8.22.3-1 illustrates the procedure for auditing service API invocation.

Pre-conditions:

1. Service API invocation logs are available at the CAPIF core function.
2. Authorization details of the AMF are available with the CAPIF core function.

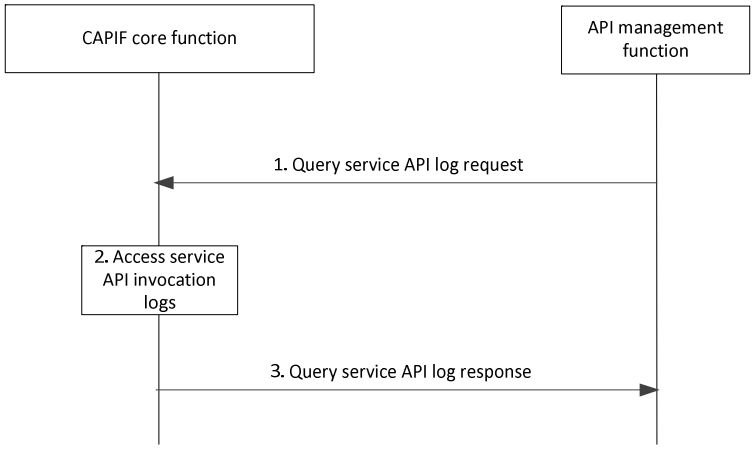

Figure 8.22.3-1: Procedure for auditing service API invocation

1. For auditing service API invocations, the API management function triggers query service API log request to the CAPIF core function.
2. Upon receiving the query service API log request, the CAPIF core function accesses the necessary service API log information for auditing purposes.
3. The CAPIF core function returns the log information to the API management function in the query service API log response.

NOTE: The API management function detecting abuse of the service API invocation and actions, subsequent to query service API log response, are out-of-scope of this specification.

8.23 CAPIF revoking API invoker authorization

8.23.1 General

The CAPIF controls the access of service API by the API invoker based on policy or usage limits. If the usage limits have exceeded, the authorization of the API invoker for accessing the service APIs is revoked. The decision to revoke the API invoker authorization may be triggered by the AEF or the CAPIF core function. The AEF can be within PLMN trust domain or within 3rd party trust domain.

In RNAA scenarios, the decision to revoke the API invoker authorization may be initiated by the CAPIF core function based on triggers at the CAPIF core function.

8.23.2 Information flows

8.23.2.1 Revoke API invoker authorization request

Table 8.23.2.1-1 describes the information flow revoke API invoker authorization request from the API exposing function to the CAPIF core function or from the CAPIF core function to the API exposing function.

Table 8.23.2.1-1: Revoke API invoker authorization request

Information element	Status	Description
API invoker identity information	M	The information that determines the identity of the API invoker
Service API identification	M	The identification information of the service API for which the authorization is revoked.
Cause	M	The cause for revoking the API invoker authorization

8.23.2.2 Revoke API invoker authorization response

Table 8.23.2.2-1 describes the information flow revoke API invoker authorization response from the CAPIF core function to the API exposing function or from the API exposing function to the CAPIF core function.

Table 8.23.2.2-1: Revoke API invoker authorization response

Information element	Status	Description
Result	M	Indicates the success or failure of revoke API invoker authorization.

8.23.2.3 Revoke API invoker authorization notify

Table 8.23.2.3-1 describes the information flow revoke API invoker authorization notify from the CAPIF core function to the API invoker.

Table 8.23.2.3-1: Revoke API invoker authorization notify

Information element	Status	Description
API invoker identity information	M	The information that determines the identity of the API invoker whose authorization has been revoked
Service API identification	M	The identification information of the service API for which the authorization is revoked.
Cause	M	The cause for revoking the API invoker authorization

8.23.3 Procedure for CAPIF revoking API invoker authorization initiated by AEF

Figure 8.23.3-1 illustrates the procedure for revoking API invoker authorization to access service API initiated by the AEF.

Pre-conditions:

1. The API invoker is authenticated and authorized to use the service API.
2. The AEF in the CAPIF is configured with the access policy to be applied to the service API invocation corresponding to the API invoker and the service API.
3. Authorization details of the AEF are available with the CAPIF core function.

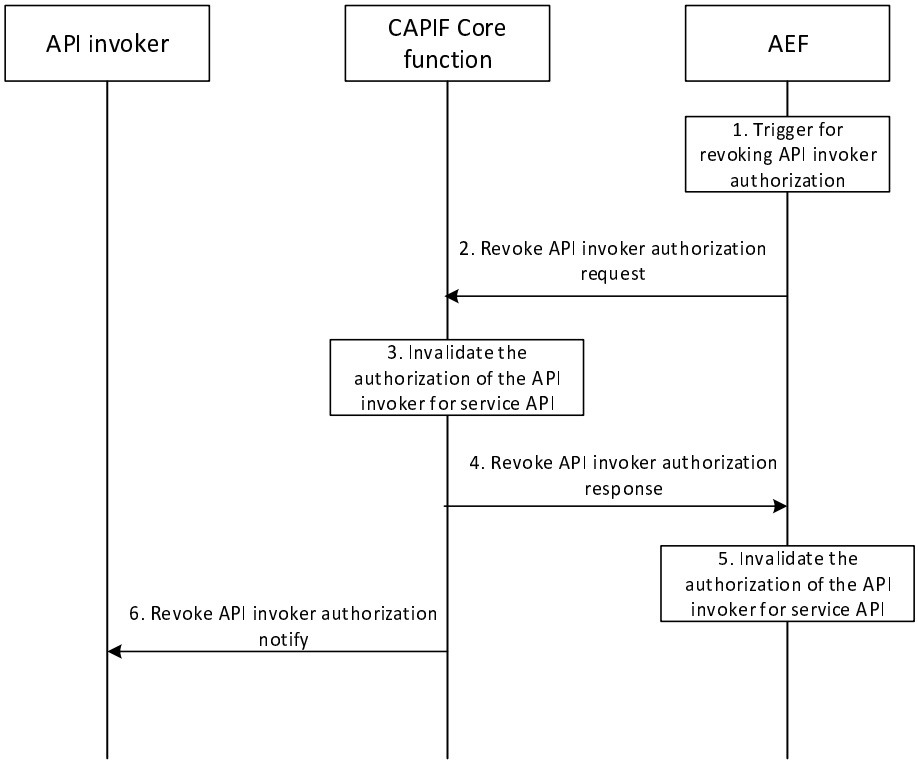

Figure 8.23.3-1: Procedure for revoking API invoker authorization initiated by AEF

1. The AEF triggers the revocation of the API invoker authorization.
2. The AEF sends revoke API invoker authorization request to the CAPIF core function with the details of the API invoker and the service API.
3. Upon receiving the information to revoke the API invoker's authorization for service API invocation, the CAPIF core function invalidates the API invoker authorization corresponding to the service API.
4. The CAPIF core function sends a revoke API invoker authorization response to the AEF.
5. Upon successful revocation of API invoker authorization corresponding to the service API at the CAPIF core function, the AEF invalidates the API invoker authorization corresponding to the service API.
6. The CAPIF core function sends a revoke API invoker authorization notify to the API invoker whose authorization to access the service API has been revoked.

8.23.4 Procedure for CAPIF revoking API invoker authorization initiated by CAPIF core function

Figure 8.23.4-1 illustrates the procedure for revoking API invoker authorization to access service API initiated by the CAPIF core function. This procedure is also used for revoking API invoker authorization supporting RNAA scenarios.

Pre-conditions:

1. The API invoker is authenticated and authorized to use the service API.
2. The AEF in the CAPIF is configured with the access policy to be applied to the service API invocation corresponding to the API invoker and the service API.

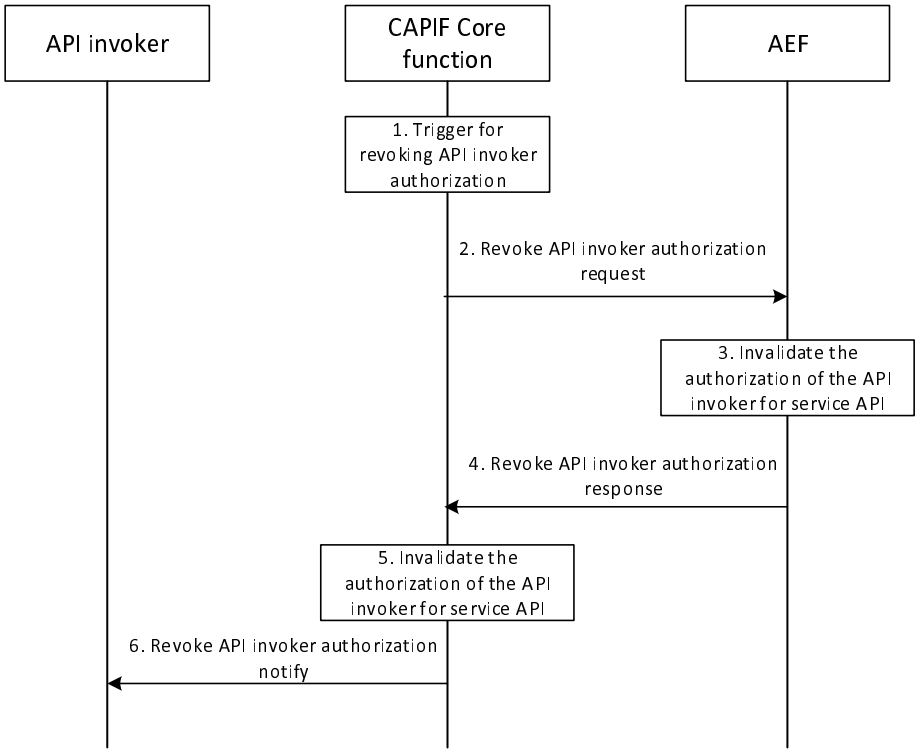

Figure 8.23.4-1: Procedure for revoking API invoker authorization initiated by CAPIF core function

1. The CAPIF core function is triggered to revoke the API invoker authorization.
2. The CAPIF core function sends revoke API invoker authorization request to the AEF with the details of the API invoker and the service API.
3. Upon receiving the information to revoke the API invoker's authorization for service API invocation, the AEF invalidates the API invoker authorization corresponding to the service API.
4. The AEF sends a revoke API invoker authorization response to the CAPIF core function.
5. The CAPIF core function invalidates the API invoker authorization corresponding to the service API.
6. The CAPIF core function sends a revoke API invoker authorization notify to the API invoker whose authorization to access the service API has been revoked.

8.24 API topology hiding management

8.24.1 General

The following procedure in this subclause corresponds to the architectural requirements on API topology hiding. The procedure in this subclause supports API topology hiding by dynamically configuring the address of the AEF providing the Service API to the AEF entry point providing the topology hiding. The API publishing function and the API exposing function can be within PLMN trust domain or within 3rd party trust domain.

8.24.2 Information flows

8.24.2.1 API topology hiding notify

Table 8.24.2.1-1 describes the information flow API topology hiding notify from the CAPIF core function to the API exposing function.

Table 8.24.2.1-1: API topology hiding notify

Information element	Status	Description
Service API identification	M	The identification information of the service API with the API topology hiding
API exposing function(s) information	M	Indicates the one or more AEF(s) which provides the service API to apply the topology hiding including the interface details (e.g. IP address, port number, URI).
Action	M	Indicates the notification action for the API topology hiding (created or revoked).

8.24.3 Procedure

Figure 8.24.3-1 illustrates the procedure for API topology hiding management by API (un)publish function.

Pre-condition:

1. Authorization details of the APF are available with the CAPIF core function.
2. The API exposing function has subscribed to CAPIF event for API topology hiding status.

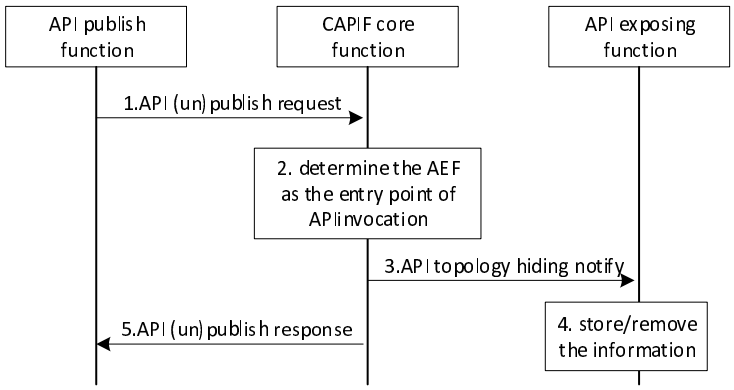

Figure 8.24.3-1: API topology hiding via API (un)publish

1. The API publishing function sends a service API publish request as described in subclause 8.3.2.1 or a service API unpublish request as described in subclause 8.4.2.1 to the CAPIF core function.
2. Upon receiving the service API (un)publish request, the CAPIF core function checks whether the API publishing function is authorized to perform the service API (un)publish. If authorized, based on the service APIs and policy:
   - For service API publish, the CCF applies the topology hiding by selecting an AEF providing the topology hiding as the entry point for service API invocation. The selected AEF information is stored with the service API information received from API publish function at the CAPIF core function (API registry).
   - For service API unpublish, the previously selected AEF as topology hinding entry point and the associated service API information at the CAPIF core function (API registry) are removed.
3. The CCF sends the API topology notify to the AEF selected as the entry point for service API invocation. The service API identification and the AEF(s) information which provides the service API details are included.
4. Upon receiving the notification, the AEF stores the received information for further service API invocation request forwarding if the action in the API topology notify indicates "created" or removes the stored API forwarding information if the action in the API topology notify indicates "revoked".
5. The CCF sends an API (un)publish response to the API publish function.

8.25 Support for CAPIF interconnection

8.25.1 General

The procedures in this subclause corresponds to the architectural requirements on CAPIF interconnection.

8.25.2 Information flows

8.25.2.1 Interconnection API publish request

Table 8.25.2.1-1 describes the information flow interconnection API publish request from CAPIF core function to CAPIF core function.

Table 8.25.2.1-1: Interconnection API publish request

Information element	Status	Description
CCF information	M	The information of the CAPIF core function which publishes APIs, may include identity, authentication and authorization information
Service API information	O (see NOTE 1)	The service API information includes the service API name, API provider name (optional), List of public IP ranges of UEs (optional and applicable only on CAPIF-6 interface), service API type, service API status (e.g. active, inactive), communication category, description, Serving Area Information (optional), AEF location (optional), interface details (e.g. IP address, port number, URI), protocols, version numbers, and data format, Service KPIs (optional).
Service API category	O (see NOTE 1)	The category of the service APIs to be published, (e.g. V2X, IoT)
Shareable information	O (see NOTE 2)	Indicates whether the service API information or the service API category can be published to other CCFs. And if sharing, a list of CAPIF provider domain information where the service API information or the service API category can be published is contained.
NOTE 1: At least one of the Service API information or Service API category shall be present. NOTE 2: If the shareable information is not present, the service API is not allowed to be shared. There is one and only one CAPIF provider domain information sharable via the CAPIF-6e interface.		

8.25.2.2 Interconnection API publish response

Table 8.25.2.2-1 describes the information flow interconnection API publish response from CAPIF core function to CAPIF core function.

Table 8.25.2.2-1: Interconnection API publish response

Information element	Status	Description
Result	M	Indicates the success or failure of publishing the service API information
Service API published information reference	O (see NOTE)	The information which can be used for referencing the information (set) about the published service API by the CCF which publishes service APIs
NOTE: This information element is included when the Result indicates success.		

8.25.2.3 Interconnection service API discover request

Table 8.25.2.3-1 describes the information flow interconnection service API discover request from one CAPIF core function to another CAPIF core function.

Table 8.25.2.3-1: Interconnection service API discover request

Information element	Status	Description
CAPIF core function identity information	M	Identity information of the CAPIF core function discovering service APIs
Query information	M	Criteria for discovering matching service APIs or CAPIF core function (e.g. service API category, Serving Area Information (optional), UE IP address (optional), preferred AEF location (optional), required API provider name (optional), interfaces, protocols, service API category, Service KPIs (optional)) (see NOTE)
NOTE: It should be possible to discover all the service APIs.		

8.25.2.4 Interconnection service API discover response

Table 8.25.2.4-1 describes the information flow interconnection service API discover response from one CAPIF core function to another CAPIF core function.

Table 8.25.2.4-1: Interconnection service API discover response

Information element	Status	Description
Result	M	Indicates the success or failure of the discovery of the service API information
Service API information	O (see NOTE)	List of service APIs corresponding to the request, including service API information as specified in Table 8.25.2.1-1.
CAPIF core function identity information	O (see NOTE)	Indicates the CAPIF core function matching the service API category in the query criteria
NOTE: The service API information or the CAPIF core function identity information or both shall be present, if the Result information element indicates that the interconnection service API discover operation is successful. Otherwise, both shall not be present.		

8.25.3 Procedure

8.25.3.1 Service API publish for CAPIF interconnection

This subclause describes the procedure for service API publish for CAPIF interoperation.

Pre-condition:

1. CCF-A and CCF-B connect to each other, and either belong to the single trust domain of the same CAPIF provider or trust domains of different CAPIF providers.
2. CCF-B is configured as the designated CAPIF core function in the trust domain of CAPIF provider A.
3. When CCF-A and CCF-B belong to trust domains of different CAPIF providers, the two CAPIF providers have business agreement for service API sharing.

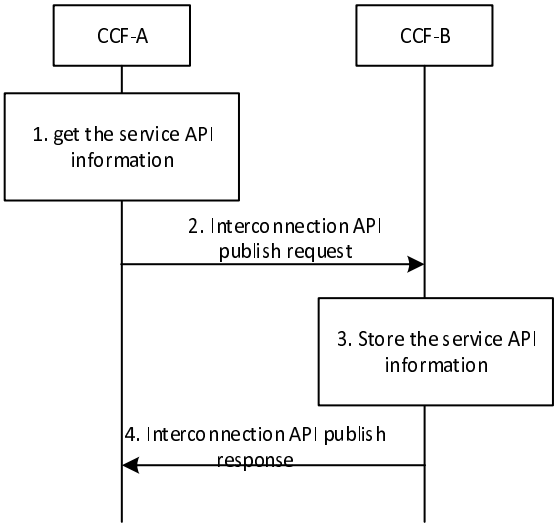

Figure 8.25.3.1-1: Interconnection API publish

1. CCF-A gets the service APIs to be shared with CCF-B from the API publish function which is in the same CAPIF provider domain of CCF-A as described in subclause 8.3.3, or from another CCF as described in this procedure.

2. Based on the shareable information for the service API or the service API category information, the CCF-A determines to publish the service API or the service API category information to the CCF-B. The CCF-A sends the interconnection API publish request to CCF-B with the details of at least one of service APIs or the category information of the service APIs, along with the identity information of CCF-A, shareable information and CAPIF provider domain information if allowed to share. The API topology hiding may be enabled.

3. CCF-B stores the service API information or service API category provided by the CCF-A.

4. CCF-B provides an interconnection API publish response to the CCF-A indicating success or failure result and triggers notifications to subscribed API invokers as described in subclause 8.8.4.

8.25.3.2 Service API discovery involving multiple CCFs

This subclause describes a procedure for service API discovery involving multiple CCFs

Pre-condition:

1. CCF-A and CCF-B connect to each other, and either belong to the single trust domain of the same CAPIF provider or trust domains of different CAPIF providers.

2. When CCF-A and CCF-B belong to trust domains of different CAPIF providers, the two CAPIF providers have business agreement for service API sharing.

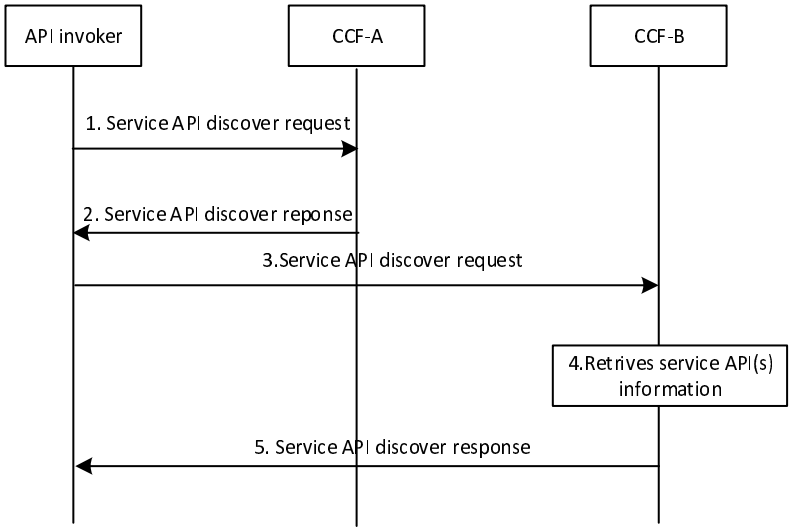

Figure 8.25.3.2-1: Service API discovery y involving multiple CCFs

1. The API invoker sends a service API discover request to the CCF-A. It includes the API invoker identity and query information.

2. The CCF-A verifies the identity of the API invoker and retrieves the stored service API(s) information and service API categories. The information of CCF-B with the service API category matching the discovery criteria is returned to API invoker in the service API discover response.

NOTE: The remaining steps are only applied when the service API category is included in the interconnection API publish request as described in subclause 8.25.2.1.

3. The API invoker sends an service API discover request to the CCF-B. The identity of API invoker is included. The query information is also provided.

4. Upon receiving the service API discover request, the CCF-B verifies the identity of the API invoker. The CCF-B retrieves the stored service API(s) information as per the query information in the service API discover request. Further, the CCF-B applies the discovery policy and performs filtering of service APIs information which matches the discovery criteria.

5. The CCF-B sends an service API discover response to the API invoker with the list of service API information for which the API invoker has the required authorization.

8.25.3.3 Service API discovery for CAPIF interconnection

This subclause describes a procedure for service API discovery for CAPIF interconnection. The CCF-A and the CCF-B may belong to the same CAPIF provider domain or different CAPIF provider domains. When the CCF-A and the CCF-B belong to different CAPIF provider domains, the two CAPIF providers shall have business agreement for service API discovery.

Pre-conditions:

1. The CCF-A is configured with the CCF-B information.

2. The CCF-B is configured with the CCF-A information.

3. The CCF-A is triggered (e.g. API invoker service API discovery, periodic service API discovery) to perform service API discovery with the CCF-B.

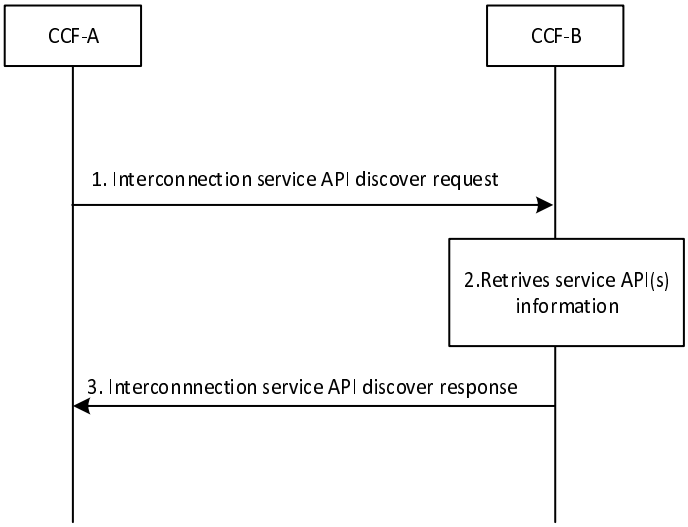

Figure 8.25.3.3-1: Service API discovery for CAPIF interconnection

1. The CCF-A sends the interconnection service API discover request to the CCF-B. The identity of the CCF-A and the query information are included.
2. The CCF-B upon receiving the interconnection service API discover request verifies the identity of the CCF-A. The CCF-B retrieves the stored service API(s) or the CCF(s) information as per the query information in the interconnection service API discover request. Further, the CCF-B applies the discovery policy and performs the filtering of service APIs or the CCF(s) information. The topology hiding policy may be applied to the retrieved list of service API information.
3. The CCF-B sends the interconnection service API discover response to the CCF-A with the list of service API information for which the CCF-A has the required authorization or the CCF(s) information that matches the discovery criteria.

8.26 Update API invoker's API list

8.26.1 General

The procedure in this subclause corresponds to the architectural requirements for updating the API invoker's API list on the CAPIF core function. The CAPIF enables API invoker to update its own API list e.g. subsequent to discovering new API(s).

8.26.2 Information flows

8.26.2.1 Update API invoker API list request

Table 8.26.2.1-1 describes the information flow update API invoker API list request from the API invoker to the CAPIF core function.

Table 8.26.2.1-1: Update API invoker API list request

Information element	Status	Description
API invoker identity information	M	Identity information of the API invoker requesting update
APIs for update	M	List of APIs that need update (e.g. enroll new API(s), disenroll API(s)).

8.26.2.2 Update API invoker API list response

Table 8.26.2.2-1 describes the information flow update API invoker API list response from the CAPIF core function to the API invoker.

Table 8.26.2.2-1: Update API invoker API list response

Information element	Status	Description
Result	M	Indicates the completely successful or partially successful or failure of the update operation
API information	O (see NOTE 1)	List of APIs and the categories of service APIs that the API invoker can access
Reason	O (see NOTE 2)	This element indicates the reason when update status is failure and for which API(s)
NOTE 1: Information element shall be present when update API invoker API list status is partial or completely successful. NOTE 2: Information element shall be present when update API invoker API list status is partial successful or failure.		

8.26.3 Procedure

Figure 8.26.3-1 illustrates the procedure for updating the API invoker API list on the CAPIF.

Pre-conditions:

1. The API invoker has been onboarded as a recognized user of the CAPIF and associated API invoker profile is provisioned.
2. The API invoker has visibility to new APIs information (e.g. updates on API catalogue or dashboard, API discovery).

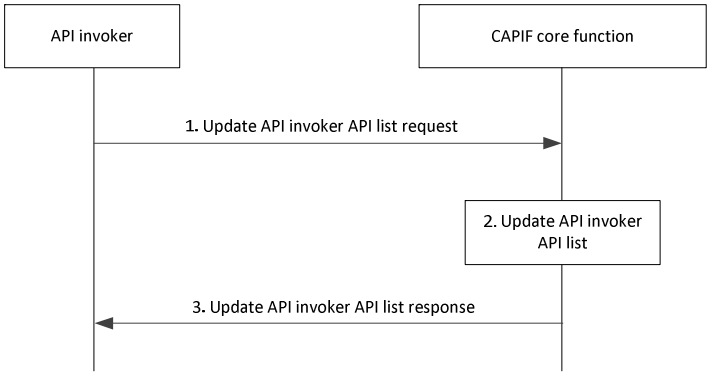

Figure 8.26.3-1: Procedure for updating the API invoker profile on the CAPIF

1. For updating of the API invoker API list on the CAPIF, the API invoker triggers update API invoker API list request towards the CAPIF core function, providing the information to be updated (e.g. enroll new APIs, disenroll APIs).
2. The CAPIF core function updates the API invoker API list of the requesting API invoker, according to the grant from the CAPIF administrator or the API management.

NOTE: Completion of updating process can require explicit grant by the CAPIF administrator or the API management, which is left out-of-scope of this solution. CAPIF can handle the grant process internally without the need of explicit grant by the CAPIF administrator.

3. The update API invoker API list response provides partial success or complete success or failure indication. Partial success and complete success result will include APIs information that the API invoker can access. When the update status is failure, the reason for failure and information for which API(s) the update operation has failed is included.

8.27 Dynamically routing service API invocation

8.27.1 General

The procedure in this subclause corresponds to the architectural requirements for dynamic routing of service API invocation. The CAPIF enables dynamically routing the service API invocation request based on the detailed information of the invocation.

8.27.2 Information flows

8.27.2.1 Obtain routing information request

Table 8.27.2.1-1 describes the information flow dynamic routing information request from the API exposing function to the CAPIF core function.

Table 8.27.2.1-1: Obtain routing information request

Information element	Status	Description
Service API identification	M	The identification information of the service API for which invocation is requested. The service API identification is part of the specific service API invocation request.
AEF identity information	M	Identity information of the entity requesting the routing information

8.27.2.2 Obtain routing information response

Table 8.27.2.2-1 describes the information flow dynamic routing information response from the CAPIF core function to the API exposing function.

Table 8.27.2.2-1: Obtain routing information response

Information element	Status	Description
Service API identification	M	The identification information of the service API for which invocation is requested.
Routing rule(s) information for service API invocation	M	Indicates the routing rule(s) for service API invocation, e.g., mapping of IP address range and AEF identity, or mapping of area serving the service API and AEF information.

8.27.3 Procedure

Figure 8.27.3-1 illustrates the procedure for dynamically routing the service API invocation from the AEF acting as service communication entry point to the destination AEF for handling service API.

Pre-conditions:

1. The API invoker has performed the service discovery and received the details of the service API which includes the information about the service communication entry point of the AEF-1 in the CAPIF.
2. The API invoker is authenticated and authorized to use the service API.
3. The AEF-1 is the AEF acting as service communication entry point for the service API, and AEF-2 is one of the multiple destination AEF which provides the service API.

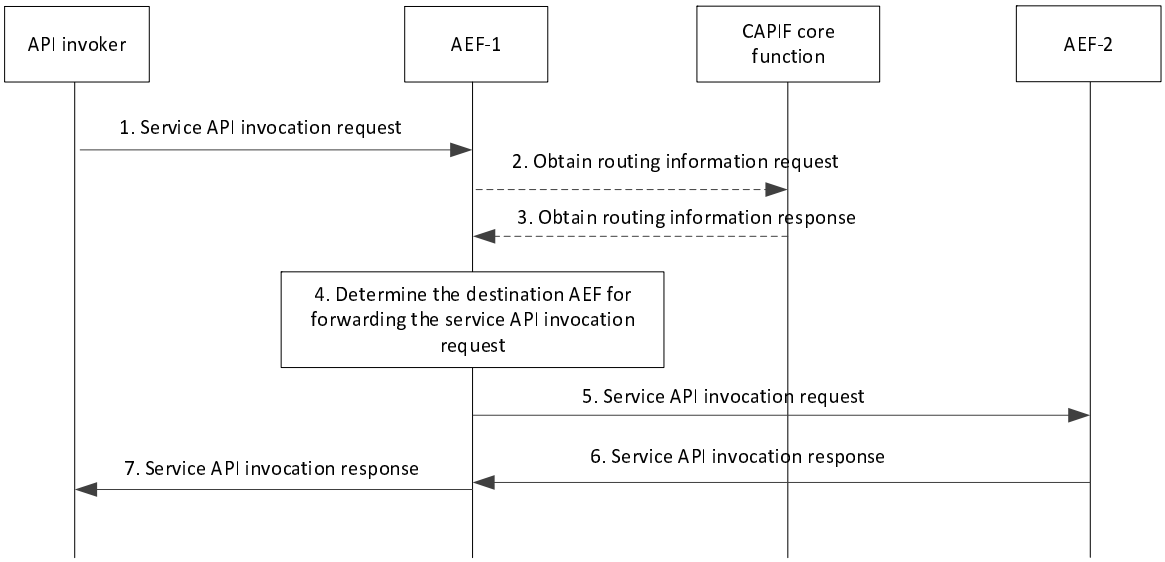

Figure 8.27.3-1: Procedure for dynamic routing of service API invocation

1. The API invoker performs service API invocation according to the interface of the service API by sending a service API invocation request towards the AEF-1 which exposes the service API towards the API invoker, and acts as topology hiding entity.
2. If the routing rule information for the service API invocation is not available, the AEF-1 sends obtain routing information request to the CAPIF core function.
3. The CAPIF core function creates routing rule information for the service API and sends obtain routing information response with the routing rule information.

NOTE: Steps 2 and 3 can be performed before step 1and after receiving the API topology hiding notify as described in subclause 8.24.3.

4. The AEF-1 further resolves the actual destination of the service API address information (AEF-2) according to the routing rule information and the invocation parameters in service API invocation request.
5. The AEF-1 forwards the incoming service API invocation request to AEF-2.
6. The AEF-2 returns the service API invocation response to AEF-1.
7. The AEF-1 sends the service API invocation response to the API invoker.

8.28 Registering the API provider domain functions on the CAPIF

8.28.1 General

The procedure in this subclause corresponds to the architectural requirements for registering the API provider domain functions on the CAPIF. This procedure registers the API provider domain functions as authorized users of the CAPIF functionalities.

Editor's Note: The security aspects of this procedure are FFS in SA3.

8.28.2 Information flows

8.28.2.1 Registration request

Table 8.28.2.1-1 describes the information flow, registration request, from the API management function to the CAPIF core function.

Table 8.28.2.1-1: Registration request

Information element	Status	Description
List of API provider domain functions	M	List of API provider domain functions including role (e.g. AEF, APF, AMF) and, if required, specific security information.
API provider name	O	The API provider name uniquely identifies an API provider (e.g. Internet Service Provider).
Security information	M	Information for CAPIF core function to validate the registration request

8.28.2.2 Registration response

Table 8.28.2.2-1 describes the information flow, registration response, from the CAPIF core function to the API management function.

Table 8.28.2.2-1: Registration response

Information element	Status	Description
List of identities	M (see NOTE 1)	List of identities, for each successfully registered API provider domain function and any specific security information.
Security information	O	Information to be used by the API provider domain function in subsequent CAPIF API invocations. Provided when registration is successful.
Reason	O (see NOTE 2)	Information related to registration result specific to individual API provider domain functions. Provided when the registration fails.
NOTE 1: Information element shall be present when at least one registration request is successful. NOTE 2: Information element may be present when at least one registration requests fail.		

8.28.3 Procedure

Figure 8.28.3-1 illustrates the procedure for registering API provider domain functions on the CAPIF core function.

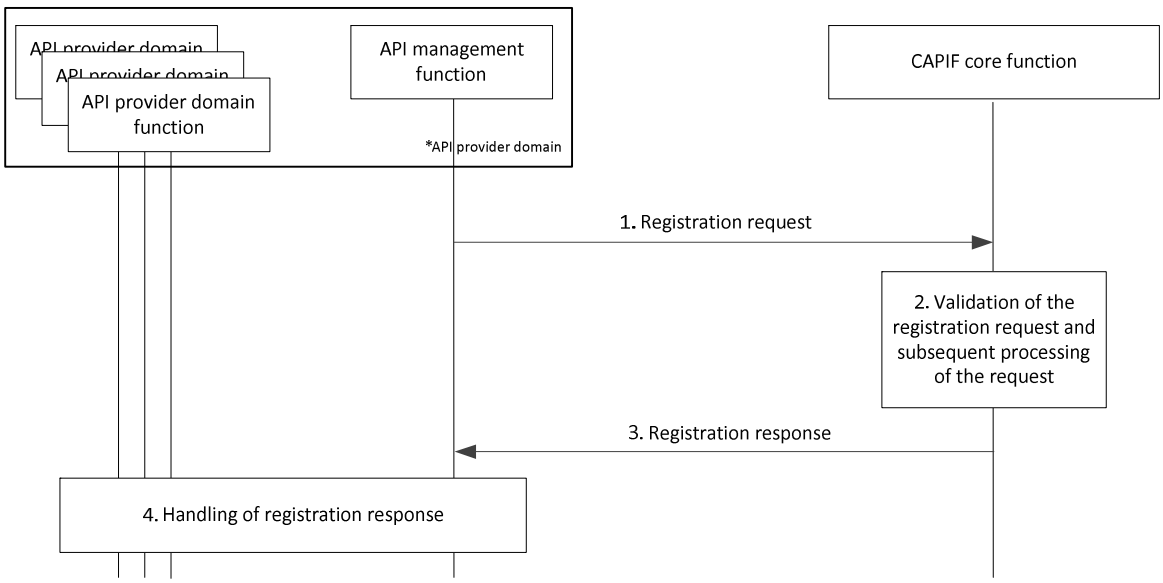

Figure 8.28.3-1: Procedure for registration of API provider domain functions on CAPIF

1. For registration of API provider domain functions on the CAPIF core function, the API management function sends a registration request to the CAPIF core function. The registration request contains a list of information about all the API provider domain functions, which require registration on the CAPIF core function.
2. The CAPIF core function validates the received request and generates the identity and other security related information for all the API provider domain functions listed in the registration request.
3. The CAPIF core function sends the generated information in the registration response message to the API management function.
4. The API management function configures the received information to the individual API provider domain functions.

8.29 Update registration information of the API provider domain functions on the CAPIF

8.29.1 General

The procedure in this subclause corresponds to the architectural requirements for update of the registration information of the API provider domain functions on the CAPIF.

Editor's Note: The security aspects of this procedure are FFS in SA3.

8.29.2 Information flows

8.29.2.1 Registration update request

Table 8.29.2.1-1 describes the information flow, registration update request, from the API management function to the CAPIF core function.

Table 8.29.2.1-1: Registration update request

Information element	Status	Description
List of API provider domain functions requiring update	M	List of API provider domain functions requiring updates, including role (e.g. AEF, APF, AMF) and, if required, specific security information.
Security information	M	Information for CAPIF core function to validate the registration request

8.29.2.2 Registration update response

Table 8.29.2.2-1 describes the information flow, registration update response, from the CAPIF core function to the API management function.

Table 8.29.2.2-1: Registration update response

Information element	Status	Description
List of identities	M (see NOTE 1)	List of identities, for each successfully updated API provider domain function and any specific security information.
Security information	O	Information to be used by the API provider domain function in subsequent CAPIF API invocations. Provided when registration update is successful.
Reason	O (see NOTE 2)	Information related to registration update result specific to individual API provider domain functions. Provided when the registration fails.
NOTE 1: Information element shall be present when at least one registration update request is successful. NOTE 2: Information element may be present when at least one registration update requests fail.		

8.29.3 Procedure

Figure 8.29.3-1 illustrates the procedure for updating the registration information of the API provider domain functions on the CAPIF core function.

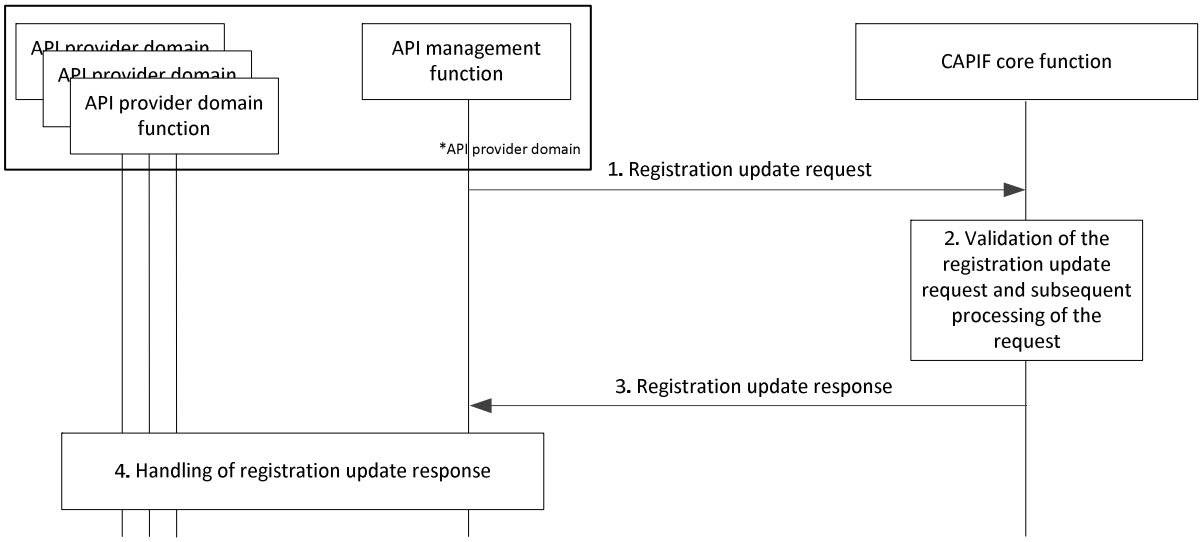

Figure 8.29.3-1: Procedure for update of registration information of API provider domain functions on CAPIF

1. For updating the registration information of API provider domain functions on the CAPIF core function, the API management function sends a registration update request to the CAPIF core function. The registration update request contains a list of information about all the API provider domain functions, which require registration update on the CAPIF core function.

2. The CAPIF core function validates the received request and updates the identity and other security related information for all the API provider domain functions listed in the registration request.

3. The CAPIF core function sends the updated information in the registration update response message to the API management function.

4. The API management function configures the received information to the individual API provider domain functions.

8.30 Deregistering the API provider domain functions on the CAPIF

8.30.1 General

The procedure in this subclause corresponds to the architectural requirements for deregistering the API provider domain functions on the CAPIF. This procedure deregisters the API provider domain functions as authorized users of the CAPIF functionalities.

Editor's Note: The security aspects of this procedure are FFS in SA3.

8.30.2 Information flows

8.30.2.1 Deregistration request

Table 8.30.2.1-1 describes the information flow, deregistration request, from the API management function to the CAPIF core function.

Table 8.30.2.1-1: Deregistration request

Information element	Status	Description
Security information	M	Information for CAPIF core function to validate the deregistration request

8.30.2.2 Deregistration response

Table 8.30.2.2-1 describes the information flow, deregistration response, from the CAPIF core function to the API management function.

Table 8.30.2.2-1: Deregistration response

Information element	Status	Description
Result	M	Indicates the success or failure of the deregistration operation

8.30.3 Procedure

Figure 8.30.3-1 illustrates the procedure for deregistering API provider domain functions on the CAPIF core function.

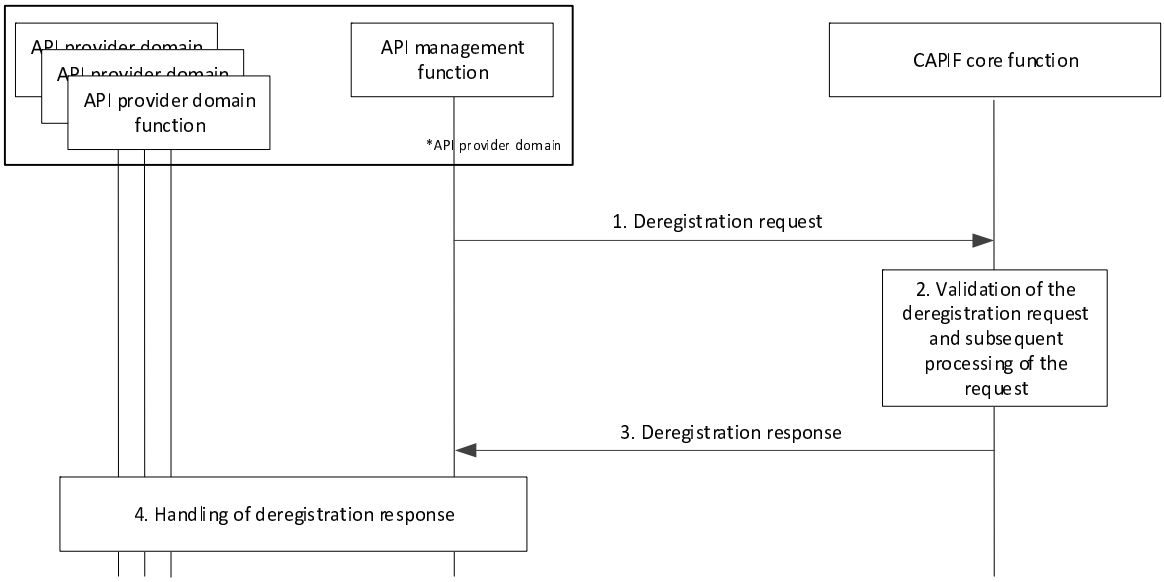

Figure 8.30.3-1: Procedure for deregistration of API provider domain functions on CAPIF

1. For deregistration of API provider domain functions on the CAPIF core function, the API management function sends a deregistration request to the CAPIF core function.
2. The CAPIF core function validates the received request and processes the deregistration request.
3. The CAPIF core function sends a deregistration response message to the API management function.
4. The API management function processes the deregistration to the individual API provider domain functions.

8.31 API invoker obtaining authorization from resource owner

8.31.1 General

CAPIF may authorize the API invoker to invoke the service API based on the authorization information from the resource owner given before the API invocation.

Clause 8.31.3 shows the procedure for obtaining the authorization information.

8.31.2 Information flows

NOTE: The security aspects of this procedure are specified in TS 33.122 [12].

8.31.3 Procedure

Figure 8.31.3-1 illustrates the procedure for API invoker obtaining authorization from resource owner.

Pre-conditions:

1. The resource owner function can communicate with the API invoker.
2. The service API access requires obtaining authorization from resource owner.

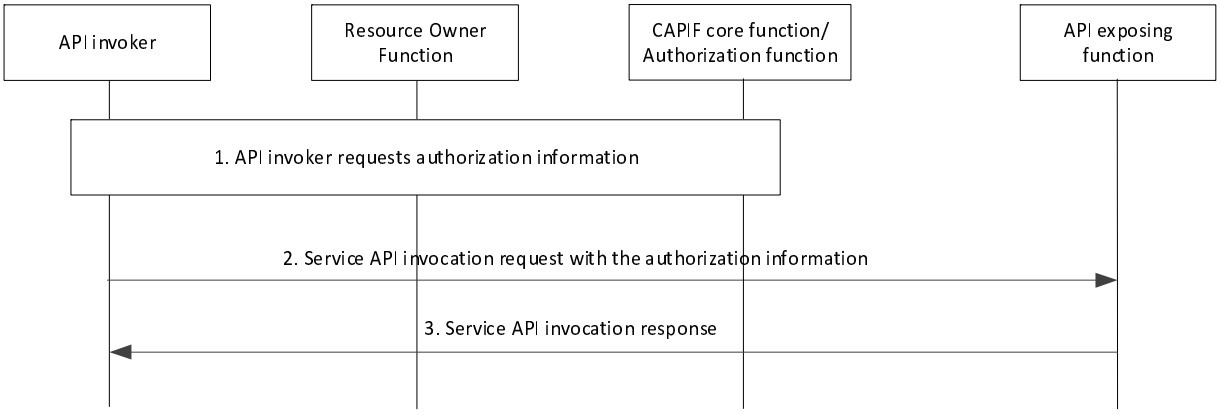

Figure 8.31.3-1: Procedure for API invoker obtaining authorization from resource owner

1. The API invoker requests to obtain resource owner authorization information to invoke the service API exposed by the API exposing function. The authorization function provides the authorization by interacting with the resource owner via the resource owner function.

NOTE: The detailed procedure to obtain the resource owner's authorization information is specified in TS 33.122 [12].

2. The API invoker sends service API invocation request to the API exposing function with the resource owner authorization information received in step 1.

3. The API invoker receives the service API invocation response resulting from the service API invocation once the API exposing function has checked whether the API invoker is authorized to invoke that service API based on the authorization information.

8.32 Reducing authorization information inquiry in a nested API invocation

8.32.1 General

The nested API invocation scenario is a scenario where an API invocation towards a first API exposing function triggers that API exposing function to request API invocation towards a second API exposing function, which is in the same API provider domain as the first API exposing function. This scenario addresses the situation in which an API may require the services of other service APIs. For example, if the API invoker invokes SEAL SS_LocationInfoRetrieval API (clause 9.4.4 of TS 23.434 [14]), the location management server (acting as an API exposing function for the API invoker and as an API invoker for the NEF) may invoke NEF API to retrieve UE location information from 5GC. In this scenario, the CAPIF may reduce the authorization information inquiries for a nested API invocation using procedure described in clause 8.32.3.

8.32.2 Information flows

NOTE: The security aspects of this procedure are specified in TS 33.122 [12].

8.32.3 Procedure

Figure 8.32.3-1 illustrates the procedure to obtain authorization information in a nested API invocation, in which an API exposing function receiving the service API invocation request interacts with another API exposing function to provide the service.

Pre-conditions:

1. The resource owner function can communicate with the API invoker.
2. The API exposing functions 1 and 2 are in the same trust domain.

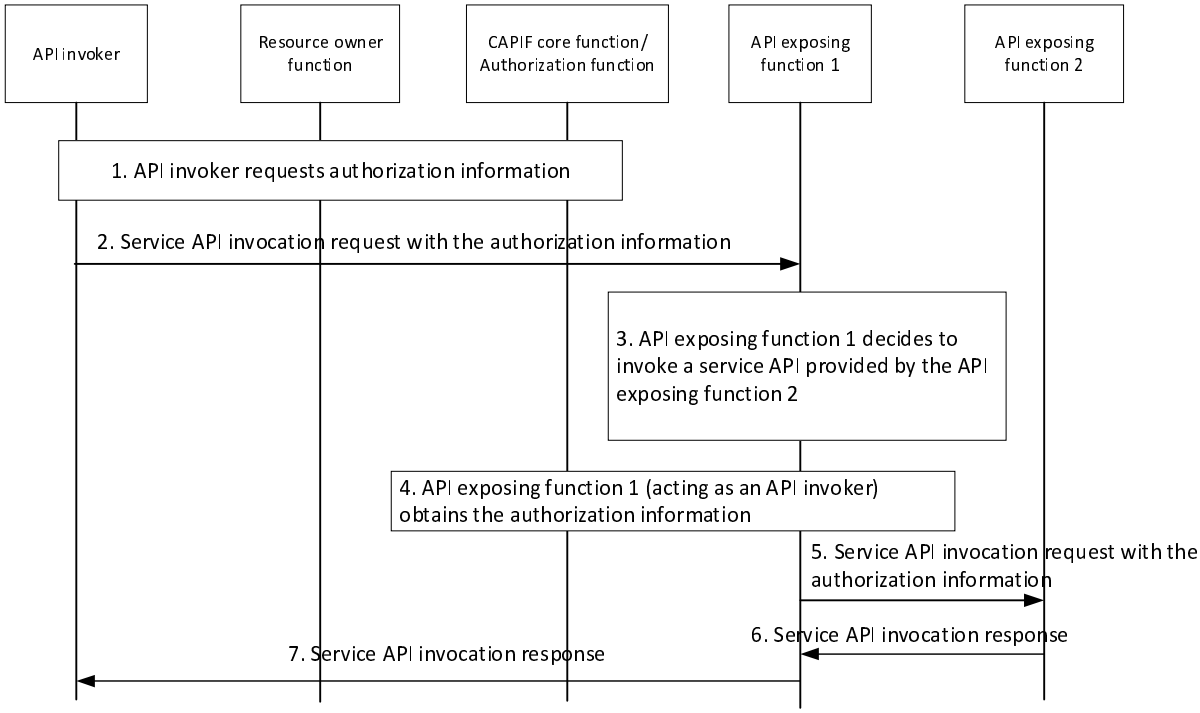

Figure 8.32.3-1: Procedure for obtaining authorization information in a nested API invocation

1. The API invoker requests authorization information to invoke the service API exposed by API exposing function 1.

NOTE 1: This step may use either the existing procedure to obtain authorization to access service API specified in clause 8.11 or the procedure that involves the resource owner function to get authorization information. For the latter case, the mechanisms to support interactions with the Resource owner are specified in 3GPP TS 33.122 [12], with further possible CAPIF support for these mechanisms being out of scope of the current release.

2. The API invoker sends a service API invocation request to API exposing function 1 with the authorization information received in step 1.
3. Based on the service API invocation request, API exposing function 1 decides to invoke another service API exposed by API exposing function 2.
4. API exposing function 1, acting as an API invoker, obtains from the CCF the authorization information to access the service API exposed by the API exposing function 2.

NOTE 2: To obtain the authorization information further interaction with the API invoker can be avoided if API exposing function 1 exchanges the authorization information provided in the API invoker's request (in step 2) with authorization information for accessing to access the service API exposed by API exposing function 2. The security mechanism involved in this exchange are out of scope of the present document.

5. API exposing function 1, acting as an API invoker sends a service API invocation request to API exposing function 2 with the authorization information received in step 4.
6. API exposing function 1 receives the service API invocation response resulting from the service API invocation once API exposing function 2 has checked whether the API invoker is authorized to invoke that service API based on the authorization information.

7. The API invoker receives the service API invocation response resulting from the service API invocation.

9 API consistency guidelines

9.1 General

This clause specifies the API consistency guidelines for all northbound APIs utilizing CAPIF architecture. The guidelines are categorized as follows:

- fundamental API guidelines, applicable to all northbound APIs utilizing CAPIF; and
- architecture design considerations, applicable to all northbound APIs utilizing CAPIF.

The API guidelines are also applicable for CAPIF APIs specified in the current specification.

9.2 Fundamental API Guidelines

The specification of each northbound API utilizing the common API framework should define:

1. the function of the API;
2. the resource(s) or endpoints involved;
3. the list of supported operations and their usage;
4. the list of input and output parameters along with applicable schemas, as required;
5. the list of supported response codes;
6. the behaviour of the network entity exposing the APIs (e.g. the CAPIF core function or the API exposing function) for each supported operation;
7. the list of applicable data types; and
8. the list of applicable protocols and data serialization formats.

In order to facilitate the consistency of the northbound APIs utilizing the common API framework it is recommended to adopt the guidelines which define the following:

1. consistent nomenclature for the operations, data structures and resources/endpoints;
2. design principles for the use of operations for common tasks; and
3. a template for the consistent documentation of APIs.

The northbound APIs utilizing the common API framework should support the following properties:

1. be extensible, such that it is possible to accommodate future requirements, including vendor-specific needs;

NOTE: The extension does not replace any existing function in Northbound APIs.

2. support access control mechanisms;
3. support charging, if applicable; and
4. be backward and forward compatible with different versions of the same API.

The guidelines above are generic with regard to the API architecture. They are valid for network APIs that follow the RESTful architectural style and that expose resources towards the API invoker, as well as for network APIs of other architectures that expose general network endpoints towards the API invoker. A network endpoint represents one end of a communication channel through which the API consumer communicates with the API producer, using messages of a

protocol defined by the API architecture. A resource is identified, and the corresponding endpoint is addressed, by a resource identifier (such as a URI).

9.3 Architecture design considerations

Northbound APIs utilizing common API framework should adhere to RESTful architecture, whenever possible. Service operations can use custom API operations (RPC-style interaction), when it is seen a better fit for the style of interaction to model, e.g. non-CRUD service operations.

NOTE: The selection of a particular API style is specific to each API implementation, and subject to Stage 3 scope.

The API design:

1. should have a uniform interface that conveys the resource/data model of the API to its client developers and:
   a. the implementation of the resource(s)/operations involved in the APIs should be hidden from the client, and adequate operations should be designed to operate on the resource(s)/data;
   b. any single API should be atomic;
   c. all resources/operations involved in APIs should be accessible through a common approach, and resources/data should be similarly modified using a consistent approach;
2. should allow the client (such as the API invoker) and the server (such as the CAPIF core function or the API exposing function) to evolve independently, i.e. the client should not have to be aware of the execution aspects of the APIs on the server;
3. should be stateless such that each request from the client (such as the API invoker) to the server (such as the CAPIF core function or the API exposing function) contains all of the information necessary for the server to understand the request;
4. should define the usage of standard operations, such as Create, Read, Update and Delete, consistently along with the applicable response codes;
5. should allow to label responses as cacheable or non-cacheable, to improve network efficiency by supporting caching in the client (such as the API invoker), if applicable in the API architecture;
6. should prevent unwanted modification of the resources/data during invocation of APIs; and
7. should support version control.

10 CAPIF core function APIs

10.1 General

Table 10.1-1 illustrates the CAPIF core function APIs.

Table 10.1-1: List of CAPIF core function APIs

API Name	API Operations	Known Consumer(s)	Communication Type
CAPIF_Discover_Service_API	Discover_Service_API	API Invoker, CAPIF core function	Request/ Response
	Subscribe_Event	API Invoker	Request/ Response
	Update_Event_Subscription	API Invoker	Request/ Response
	Notify_Event	API Invoker	Notify
	Unsubscribe_Event	API Invoker	Request/ Response
CAPIF_Publish_Service_API	Publish_Service_API	API Publishing Function, CAPIF core function	Request/ Response
	Unpublish_Service_API	API Publishing Function, CAPIF core function	Request/ Response
	Update_Service_API	API Publishing Function, CAPIF core function	Request/ Response
	Get_Service_API	API Publishing Function, CAPIF core function	Request/ Response
	Subscribe_Event	API Publishing Function	Request/ Response
	Update_Event_Subscription	API Publishing Functionr	Request/ Response
	Notify_Event	API Publishing Function	Notify
	Unsubscribe_Event	API Publishing Function	Request/ Response
CAPIF_Events API	Subscribe_Event	API Invoker, API Publishing Function, API Management Function, API Exposing Function	Request/ Response
	Update_Event_Subscription	API Invoker, API Publishing Function, API Management Function, API Exposing Function	Request/ Response
	Notify_Event	API Invoker, API Publishing Function, API Management Function, API Exposing Function	Notify
	Unsubscribe_Event	API Invoker, API Publishing Function, API Management Function, API Exposing Function	Request/ Response
CAPIF_API_Invoker_management API	Onboard_API_Invoker	API Invoker	Request/ Response
	Offboard_API_Invoker	API Invoker	Request/ Response
	Subscribe_Event	API Management Function	Request/ Response
	Update_Event_Subscription	API Management Function	Request/ Response
	Notify_Event	API Management Function	Notify
	Unsubscribe_Event	API Management Function	Request/ Response
CAPIF_API_Provider_Management _API	Register_API_Provider	API Management Function	Request/Response
	Update_API_Provider	API Management Function	Request/Response
	Deregister_API_Provider	API Management Function	Request/Response
CAPIF_Security API	Obtain_Security_Method	API Invoker	Request/ Response
	Obtain_Authorization	API Invoker	Request/ Response
	Obtain_API_Invoker_Info	API Exposing Function	Request/ Response
	Revoke_Authorization	API Exposing Function	Request/ Response
CAPIF_Monitoring API	Subscribe_Event	API Management Function	Request/ Response
	Update_Event_Subscription	API Management Function	Request/ Response
	Notify_Monitoring_Service_ Event	API Management Function	Notify
	Unsubscribe_Event	API Management Function	Request/ Response
CAPIF_Logging_API_Invocation API	Log_API_Invocation	API exposing function	Request/ Response
CAPIF_Auditing API	Query_API_Invocation_Log	API management function	Request/ Response

CAPIF_Access_Control_Policy API	Obtain_Access_Control_Policy	API exposing function	Request/Response
CAPIF_Routing_Info API	Obtain_Routing_Info	API exposing function	Request/Response

10.2 CAPIF_Discover_Service_API API

10.2.1 General

API description: This API enables the API invoker to communicate with the CAPIF core function to discover the published service API information over CAPIF-1 or CAPIF-1e.

10.2.2 Discover_Service_API operation

API operation name: Discover_Service_API

Description: Provides the published service APIs information.

Known Consumers: API invoker.

Inputs: Refer subclause 8.7.2.1.

Outputs: Refer subclause 8.7.2.2.

See subclause 8.7.3 for the details of usage of this API operation.

10.2.3 Subscribe_Event operation

API operation name: Subscribe_Event

Description: Provides subscription to the CAPIF related event information.

Known Consumers: API invoker.

Inputs: Refer subclause 8.8.2.1.

Outputs: Refer subclause 8.8.2.2.

See subclause 8.8.3 for the details of usage of this API operation.

10.2.4 Notify_Event operation

API operation name: Notify_Event

Description: Provides the relevant CAPIF event information to the subscribed entities.

Known Consumers: API invoker.

Inputs: Refer subclause 8.8.2.3.

Outputs: Refer subclause 8.8.2.4.

See subclause 8.8.4 for the details of usage of this API operation.

10.2.5 Unsubscribe_Event operation

API operation name: Unsubscribe_Event

Description: Unsubscription to the CAPIF event information.

Known Consumers: API invoker.

Inputs: Refer subclause 8.8.2.5.

Outputs: Refer subclause 8.8.2.6.

See subclause 8.8.5 for the details of usage of this API operation.

10.2.6 Update_Event_Subscription operation

API operation name: Update_Event_Subscription

Description: Updates a subscription to CAPIF related event information.

Known Consumers: API invoker.

Inputs: Refer subclause 8.8.2.7.

Outputs: Refer subclause 8.8.2.8.

See subclause 8.8.5a for the details of usage of this API operation.

10.3 CAPIF_Publish_Service_API API

10.3.1 General

API description: This API enables the API publishing function to communicate with the CAPIF core function to publish the service API information and manage the published service API information over CAPIF-4.

NOTE: Stage 3 can decide whether the API for CAPIF_Publish_Service_API can be enabled over CAPIF-4.

10.3.2 Publish_Service_API operation

API operation name: Publish_Service_API

Description: Publish the service API information.

Known Consumers: API publishing function.

Inputs: Refer subclause 8.3.2.1.

Outputs: Refer subclause 8.3.2.2.

See subclause 8.3.3 for the details of usage of this API operation.

10.3.3 Unpublish_Service_API operation

API operation name: Unpublish_Service_API

Description: Remove the published service API information.

Known Consumers: API publishing function.

Inputs: Refer subclause 8.4.2.1.

Outputs: Refer subclause 8.4.2.2.

See subclause 8.4.3 for the details of usage of this API operation.

10.3.4 Update_Service_API operation

API operation name: Update_Service_API

Description: Update the published service API information.

Known Consumers: API publishing function.

Inputs: Refer subclause 8.6.2.1.

Outputs: Refer subclause 8.6.2.2.

See subclause 8.6.3 for the details of usage of this API operation.

10.3.5 Get_Service_API operation

API operation name: Get_Service_API

Description: Retrieve the published service API information.

Known Consumers: API publishing function.

Inputs: Refer subclause 8.5.2.1.

Outputs: Refer subclause 8.5.2.2.

See subclause 8.5.3 for the details of usage of this API operation.

10.3.6 Subscribe_Event operation

API operation name: Subscribe_Event

Description: Provides subscription to the CAPIF related event information.

Known Consumers: API publishing function.

Inputs: Refer subclause 8.8.2.1.

Outputs: Refer subclause 8.8.2.2.

See subclause 8.8.3 for the details of usage of this API operation.

10.3.7 Notify_Event operation

API operation name: Notify_Event

Description: Provides the relevant CAPIF event information to the subscribed entities.

Known Consumers: API publishing function.

Inputs: Refer subclause 8.8.2.3.

Outputs: Refer subclause 8.8.2.4.

See subclause 8.8.4 for the details of usage of this API operation.

10.3.8 Unsubscribe_Event operation

API operation name: Unsubscribe_Event

Description: Unsubscription to the CAPIF event information.

Known Consumers: API publishing function.

Inputs: Refer subclause 8.8.2.5.

Outputs: Refer subclause 8.8.2.6.

See subclause 8.8.5 for the details of usage of this API operation.

10.3.9 Update_Event_Subscription operation

API operation name: Update_Event_Subscription

Description: Updates a subscription to CAPIF related event information.

Known Consumers: API publishing function.

Inputs: Refer subclause 8.8.2.7.

Outputs: Refer subclause 8.8.2.8.

See subclause 8.8.5a for the details of usage of this API operation.

10.4 CAPIF_Events API

10.4.1 General

API description: This API enables the API subscribing entity to communicate with the CAPIF core function to subscribe to and unsubscribe from CAPIF events and receive subsequent notification of CAPIF events. This API is used for the subscription to and notifications of those CAPIF events that are not bound to any of the other CAPIF core function APIs. The following are the key functionalities:

- API invoker subscribes to CAPIF events over CAPIF-1 or CAPIF-1e.
- API invoker receives notifications for subscribed CAPIF events over CAPIF-1 or CAPIF-1e.
- API invoker unsubscribes from CAPIF events over CAPIF-1 or CAPIF-1e.
- API invoker updates subscriptions for CAPIF events over CAPIF-1 or CAPIF-1e.
- API exposing function subscribes to CAPIF events over CAPIF-3.
- API exposing function receives notifications for subscribed CAPIF events over CAPIF-3.
- API exposing function unsubscribes from CAPIF events over CAPIF-3.
- API exposing function updates subscriptions for CAPIF events over CAPIF-3.
- API publishing function subscribes to CAPIF events over CAPIF-4.
- API publishing function receives notifications for subscribed CAPIF events over CAPIF-4.
- API publishing function unsubscribes from CAPIF events over CAPIF-4.
- API publishing function updates subscriptions for CAPIF events over CAPIF-4.
- API management function subscribes to CAPIF events over CAPIF-5.
- API management function receives notifications for subscribed CAPIF events over CAPIF-5.
- API management function unsubscribes from CAPIF events over CAPIF-5.
- API management function updates subscriptions for CAPIF events over CAPIF-5.

NOTE: Stage 3 can further decide if CAPIF_Events API can be further fine grained into more APIs.

10.4.2 Subscribe_Event operation

API operation name: Subscribe_Event

Description: Provides subscription to the CAPIF related event information.

Known Consumers: API invoker, API publishing function, API management function, API exposing function.

Inputs: Refer subclause 8.8.2.1.

Outputs: Refer subclause 8.8.2.2.

See subclause 8.8.3 for the details of usage of this API operation.

10.4.3 Notify_Event operation

API operation name: Notify_Event

Description: Provides the relevant CAPIF event information to the subscribed entities.

Known Consumers: API invoker, API publishing function, API management function, API exposing function.

Inputs: Refer subclause 8.8.2.3.

Outputs: Refer subclause 8.8.2.4.

See subclause 8.8.4 for the details of usage of this API operation.

10.4.4 Unsubscribe_Event operation

API operation name: Unsubscribe_Event

Description: Unsubscription to the CAPIF event information.

Known Consumers: API invoker, API publishing function, API management function, API exposing function.

Inputs: Refer subclause 8.8.2.5.

Outputs: Refer subclause 8.8.2.6.

See subclause 8.8.5 for the details of usage of this API operation.

10.4.5 Update_Event_Subscription operation

API operation name: Update_Event_Subscription

Description: Updates a subscription to CAPIF related event information.

Known Consumers: API invoker, API publishing function, API management function, API exposing function.

Inputs: Refer subclause 8.8.2.7.

Outputs: Refer subclause 8.8.2.8.

See subclause 8.8.5a for the details of usage of this API operation.

10.5 CAPIF_API_invoker_management API

10.5.1 General

API description: This API enables the API invoker to communicate with the CAPIF core function to enroll as a registered user of CAPIF and manage the enrollment information over CAPIF-1 or CAPIF-1e.

10.5.2 Onboard_API_Invoker operation

API operation name: Onboard_API_Invoker

Description: Enrolls the API invoker as a recognized user of the CAPIF.

Known Consumers: API invoker.

Inputs: Refer subclause 8.1.2.1.

Outputs: Refer subclause 8.1.2.2.

See subclause 8.1.3 for the details of usage of this API operation.

10.5.3 Offboard_API_Invoker operation

API operation name: Offboard_API_Invoker

Description: Cancels enrollment of the API invoker as a recognized user of the CAPIF.

Known Consumers: API invoker.

Inputs: Refer subclause 8.2.2.1.

Outputs: Refer subclause 8.2.2.2.

See subclause 8.2.3 for the details of usage of this API operation.

10.5.4 Subscribe_Event operation

API operation name: Subscribe_Event

Description: Provides subscription to the CAPIF related event information.

Known Consumers: API management function.

Inputs: Refer subclause 8.8.2.1.

Outputs: Refer subclause 8.8.2.2.

See subclause 8.8.3 for the details of usage of this API operation.

10.5.5 Notify_Event operation

API operation name: Notify_Event

Description: Provides the relevant CAPIF event information to the subscribed entities.

Known Consumers: API management function.

Inputs: Refer subclause 8.8.2.3.

Outputs: Refer subclause 8.8.2.4.

See subclause 8.8.4 for the details of usage of this API operation.

10.5.6 Unsubscribe_Event operation

API operation name: Unsubscribe_Event

Description: Unsubscription to the CAPIF event information.

Known Consumers: API management function.

Inputs: Refer subclause 8.8.2.5.

Outputs: Refer subclause 8.8.2.6.

See subclause 8.8.5 for the details of usage of this API operation.

10.5.7 Update_Event_Subscription operation

API operation name: Update_Event_Subscription

Description: Updates a subscription to CAPIF related event information.

Known Consumers: API management function.

Inputs: Refer subclause 8.8.2.7.

Outputs: Refer subclause 8.8.2.8.

See subclause 8.8.5a for the details of usage of this API operation.

10.6 CAPIF_Security API

10.6.1 General

API description: This API enables the API invoker to communicate with the CAPIF core function to authenticate and obtain authorization to access service APIs over CAPIF-1 or CAPIF-1e. This API also enables the API exposing function (AEF) to obtain API invoker information and revoke API invoker authorization over CAPIF-3.

10.6.2 Obtain_Security_Method operation

API operation name: Obtain_Security_Method

Description: Obtain information about service API security method with CAPIF core function for service API invocations.

Known Consumers: API invoker.

Inputs: Refer subclause 8.10.2.

Outputs: Refer subclause 8.10.2.

See subclause 8.10.3 for the details of usage of this API operation.

10.6.3 Obtain_Authorization operation

API operation name: Obtain_Authorization

Description: Provides the authorization information to access relevant service API.

Known Consumers: API invoker.

Inputs: Refer subclause 8.11.2.

Outputs: Refer subclause 8.11.2.

See subclause 8.11.3 for the details of usage of this API operation.

10.6.4 Obtain_API_Invoker_Info operation

API operation name: Obtain_API_Invoker_Info

Description: Obtains the API invoker information.

Known Consumers: API exposing function.

Inputs: Refer subclause 8.16.2.1.

Outputs: Refer subclause 8.16.2.2.

See subclause 8.16.3 for the details of usage of this API operation.

10.6.5 Revoke_Authorization operation

API operation name: Revoke_Authorization

Description: Revokes API invoker authorization to access service API.

Known Consumers: API exposing function.

Inputs: Refer subclause 8.23.2.

Outputs: Refer subclause 8.23.2.

See subclause 8.23.3 for the details of usage of this API operation.

10.7 CAPIF_Monitoring API

10.7.1 General

API description: This API enables the API management function to communicate with the CAPIF core function to subscribe to and unsubscribe from CAPIF events related to monitoring and receive subsequent notification of CAPIF monitoring events over CAPIF-5.

NOTE: Stage 3 can decide whether the API for CAPIF_Monitoring can be enabled over CAPIF-5.

10.7.2 Subscribe_Event operation

API operation name: Subscribe_Event

Description: Provides subscription to the CAPIF related event information.

Known Consumers: API management function.

Inputs: Refer subclause 8.8.2.1.

Outputs: Refer subclause 8.8.2.2.

See subclause 8.8.3 for the details of usage of this API operation.

10.7.3 Notify_Monitoring_Service_Event operation

API operation name: Notify_Monitoring_Service_Event

Description: Provides the notification of the events related to monitoring service API invocations to the subscribed API management function.

Known Consumers: API management function.

Inputs: Refer subclause 8.21.2.1.

Outputs: Refer subclause 8.21.2.2.

See subclause 8.21.3 for the details of usage of this API operation.

10.7.4 Unsubscribe_Event operation

API operation name: Unsubscribe_Event

Description: Unsubscription to the CAPIF event information.

Known Consumers: API management function.

Inputs: Refer subclause 8.8.2.5.

Outputs: Refer subclause 8.8.2.6.

See subclause 8.8.5 for the details of usage of this API operation.

10.7.5 Update_Event_Subscription operation

API operation name: Update_Event_Subscription

Description: Updates a subscription to CAPIF related event information.

Known Consumers: API management function.

Inputs: Refer subclause 8.8.2.7.

Outputs: Refer subclause 8.8.2.8.

See subclause 8.8.5a for the details of usage of this API operation.

10.8 CAPIF_Logging_API_Invocation API

10.8.1 General

API description: This API enables the API exposing function to communicate with the CAPIF core function to log the information related to service API invocation over CAPIF-3.

NOTE: Stage 3 can decide whether the API for CAPIF_Logging_API_Invocation can be enabled over CAPIF-3.

10.8.2 Log_API_Invocation operation

API operation name: Log_API_Invocation

Description: Enables to log API invocation information.

Known Consumers: API exposing function.

Inputs: Refer subclause 8.19.2.1.

Outputs: Refer subclause 8.19.2.2.

See subclause 8.19.3 for the details of usage of this API operation.

10.9 CAPIF_Auditing API

10.9.1 General

API description: This API enables the API management function to communicate with the CAPIF core function to retrieve the log information related to service API invocation over CAPIF-5.

NOTE: Stage 3 can decide whether the API for CAPIF_Auditing can be enabled over CAPIF-5.

10.9.2 Query_ API_Invocation_Log operation

API operation name: Query_API_Invocation_Log

Description: Query the API invocation log information.

Known Consumers: API management function.

Inputs: Refer subclause 8.22.2.1.

Outputs: Refer subclause 8.22.2.2.

See subclause 8.22.3 for the details of usage of this API operation.

10.10 CAPIF_Access_Control_Policy API

10.10.1 General

API description: This API enables the API exposing function to obtain the policy to perform access control on the service API invocations.

10.10.2 Obtain_Access_Control_Policy operation

API operation name: Obtain_Access_Control_Policy

Description: Allows obtaining the policy to perform access control on the service API invocations.

Known Consumers: API exposing function.

Inputs: Refer subclause 8.12.2.1.

Outputs: Refer subclause 8.12.2.2.

See subclause 8.12.3 for the details of usage of this API operation.

10.11 CAPIF_Routing_Info API

10.11.1 General

API description: This API enables the API exposing function to obtain the routing information to forward the API invocation to another API exposing function.

10.11.2 Obtain_Routing_Info operation

API operation name: Obtain_Routing_Info

Description: Allows obtaining the API routing information.

Known Consumers: API exposing function.

Inputs: Refer subclause 8.27.2.1.

Outputs: Refer subclause 8.27.2.2.

See subclause 8.27.3 for the details of usage of this API operation.

10.12 CAPIF_API_provider_management API

10.12.1 General

API description: This API enables the API Management Function to communicate with the CAPIF core function to register the API provider domain functions as authorized users of the CAPIF functionalities.

10.12.2 Register_API_Provider operation

API operation name: Register_API_Provider

Description: Registers the API provider domain functions as authorized users of the CAPIF.

Known Consumers: API Management Function.

Inputs: Refer subclause 8.28.2.1.

Outputs: Refer subclause 8.28.2.2.

See subclause 8.28.3 for the details of usage of this API operation.

10.12.3 Update_API_Provider operation

API operation name: Update_API_Provider

Description: Updates registration information of the API provider domain functions.

Known Consumers: API Management Function.

Inputs: Refer subclause 8.29.2.1.

Outputs: Refer subclause 8.29.2.2.

See subclause 8.29.3 for the details of usage of this API operation.

10.12.4 Deregister_API_Provider operation

API operation name: Deregister_API_Provider

Description: De-registers the API provider domain functions on the CAPIF core function.

Known Consumers: API Management Function.

Inputs: Refer subclause 8.30.2.1.

Outputs: Refer subclause 8.30.2.2.

See subclause 8.30.3 for the details of usage of this API operation.

11 API exposing function APIs

11.1 General

Table 11.1-1 illustrates the API exposing function APIs.

Table 11.1-1: List of API exposing function APIs

API Name	API Operations	Known Consumer(s)	Communication Type
AEF_Security API	Revoke_Authorization	CAPIF Core Function	Request/ Response
	Initiate_Authentication	API Invoker	Request/ Response

11.2 AEF_Security API

11.2.1 General

API description: This API allows CAPIF core function to revoke access to service APIs and API invokers to request the authentication parameters necessary for authentication of the API invoker available with the API exposing function.

11.2.2 Revoke_Authorization operation

API operation name: Revoke_Authorization

Description: Revokes API invoker authorization to access service API.

Known Consumers: CAPIF core function.

Inputs: Refer subclause 8.23.2.

Outputs: Refer subclause 8.23.2.

See subclause 8.23.4 for the details of usage of this API operation.

11.2.3 Initiate_Authentication operation

API operation name: Initiate_Authentication

Description: Authentication between the API invoker and the AEF prior to service API invocation.

Known Consumers: API Invoker.

Inputs: Refer subclause 8.14.2.

Outputs: Refer subclause 8.14.2.

See subclause 8.14.3 for the details of usage of this API operation.

Annex A (informative): Overview of CAPIF operations

Depicted in figure A-1 is the overview of CAPIF operations. CAPIF operations occur between different actors involving the API invoker, the CAPIF core function, the API exposing function, the API publishing function and the API management function, and optionally the resource owner function for RNAA. High level CAPIF interactions between the actors are shown in figure A-1. This figure is only provided for illustration purposes, and does not represent the order of operations.

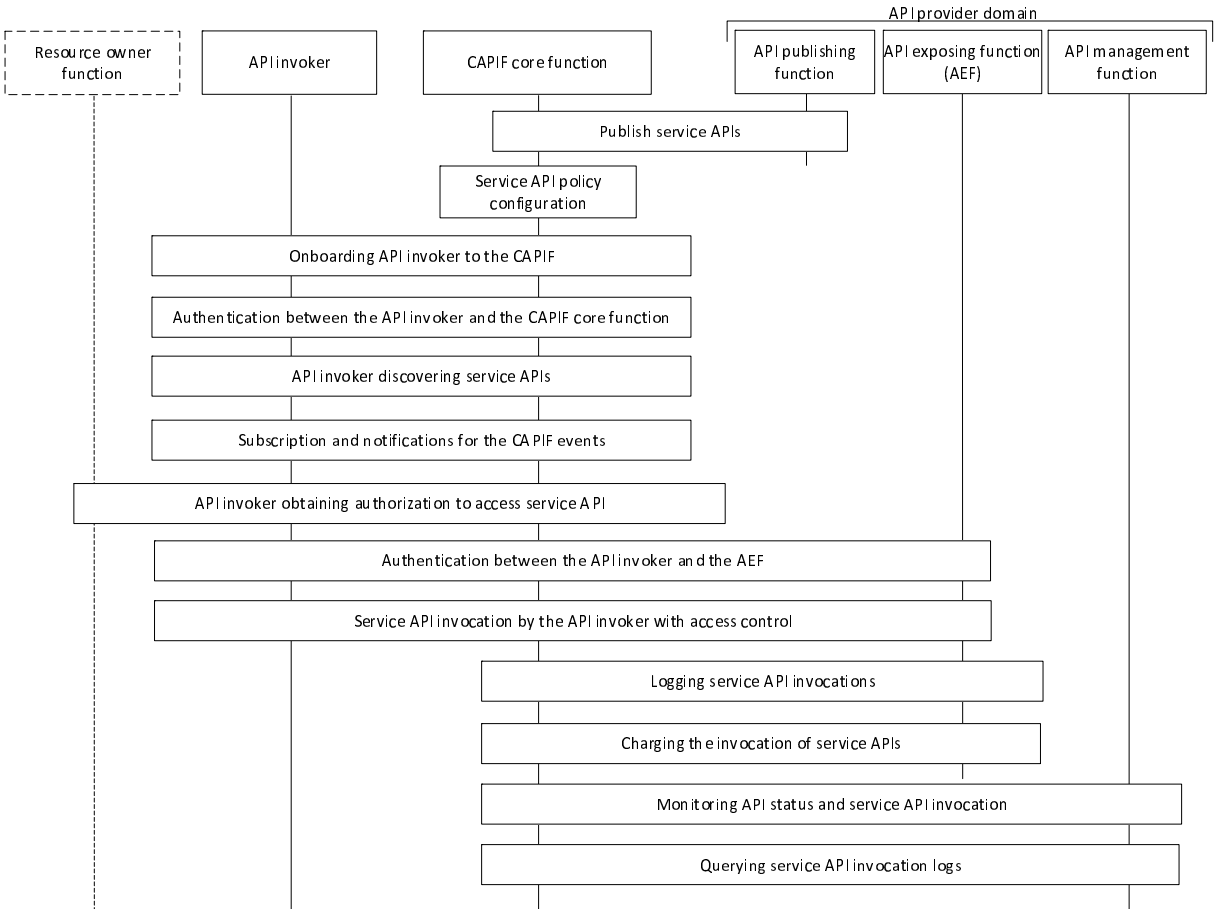

Figure A-1: Overview of CAPIF operations

The CAPIF defines the functional entities in subclause 6.3.

The CAPIF defines the reference points between the functional entities in subclause 6.4.

The following operations require the communication between the CAPIF entities:

1. Publishing service APIs: the API provider utilizes the API publishing function over CAPIF-4 reference point to publish the service APIs on the CAPIF core function, as specified in subclause 8.3 of this specification;

2. Discovering service APIs: the API invoker discovers the service APIs over CAPIF-1/CAPIF-1e reference points, as specified in subclause 8.7 of this specification;

3. API event subscription and notification: the API invoker subscribes to and receive service API event notifications over CAPIF-1/CAPIF-1e reference points, as specified in subclause 8.8 of this specification;

4. Authenticating with CAPIF: the API invoker authenticates itself over CAPIF-1/CAPIF-1e reference points, as specified in subclause 8.10 of this specification;

5. Authorizing with CAPIF: the API invoker obtains service API authorization over CAPIF-1/CAPIF-1e reference points, as specified in subclause 8.11 of this specification. In RNAA scenarios, API authorization is based on the authorization information obtained from the resource owner, as specified in clause 8.31;

6. Topology hiding: the API provider, to hide the topology, utilizes the API exposing function over CAPIF-3 reference point, as specified in subclause 8.13 of this specification;

7. Authenticating the API invoker prior to service API invocation: the API provider, to authenticate the API invoker prior to the service API invocation, utilizes the API exposing function over CAPIF-2/CAPIF-2e and CAPIF-3, as specified in subclause 8.14 of this specification;

8. Authenticating the API invoker upon the service API invocation: the API provider, to authenticate the API invoker upon invocation of the service APIs, utilizes the API exposing function over CAPIF-2/CAPIF-2e and CAPIF-3, as specified in subclause 8.15 of this specification;

9. Authorizing API invoker: the API provider, to authorize the API invoker to access the service APIs, utilizes the API exposing function over CAPIF-2/CAPIF-2e and CAPIF-3, as specified in subclause 8.16 of this specification;

10. Access control: the API provider, to control the access of the service API by the API invoker based on policy or usage limits,

    - utilizes the API exposing function over CAPIF-2/CAPIF-2e and CAPIF-3, as specified in subclause 8.17 of this specification; or

    - in a cascaded deployment, utilizes API exposing functions over CAPIF-2/CAPIF-2e, as specified in subclause 8.18 of this specification;

11. Logging service: the API provider, to maintain the log of the API invocations at the CAPIF core function for services such as charging, invocation history, utilizes the API exposing function over CAPIF-3, as specified in subclause 8.19 of this specification;

12. Charging service: the API provider, to facilitate charging of the API invocations, utilizes the API exposing function over CAPIF-3, as specified in subclause 8.20 of this specification;

13. Service monitoring: the API provider, to facilitate monitoring such as API invoker's ID and IP address, utilizes the API management function over CAPIF-5, as specified in subclause 8.21 of this specification; and

14. Auditing: the API provider, for auditing, utilizes the API management function over CAPIF-5, as specified in subclause 8.22 of this specification.

Annex B (informative): CAPIF relationship with network exposure aspects of 3GPP systems

This annex provides the relationship of CAPIF with network exposure aspects of 3GPP systems. Any system exposing capabilities as service APIs can implement CAPIF. Generic model for CAPIF utilization by service API provider is included. Network exposure aspects of EPS and 5GS are considered for illustration.

NOTE: As there are no impacts on CAPIF's relationship with network exposure aspects of 3GPP systems due to deployment of 3rd party trust domain, it is not illustrated in the figures.

B.0 CAPIF utilization by service API provider

Figure B.0-1 illustrates the service API interaction with the CAPIF for utilizing framework aspects provided by the CAPIF.

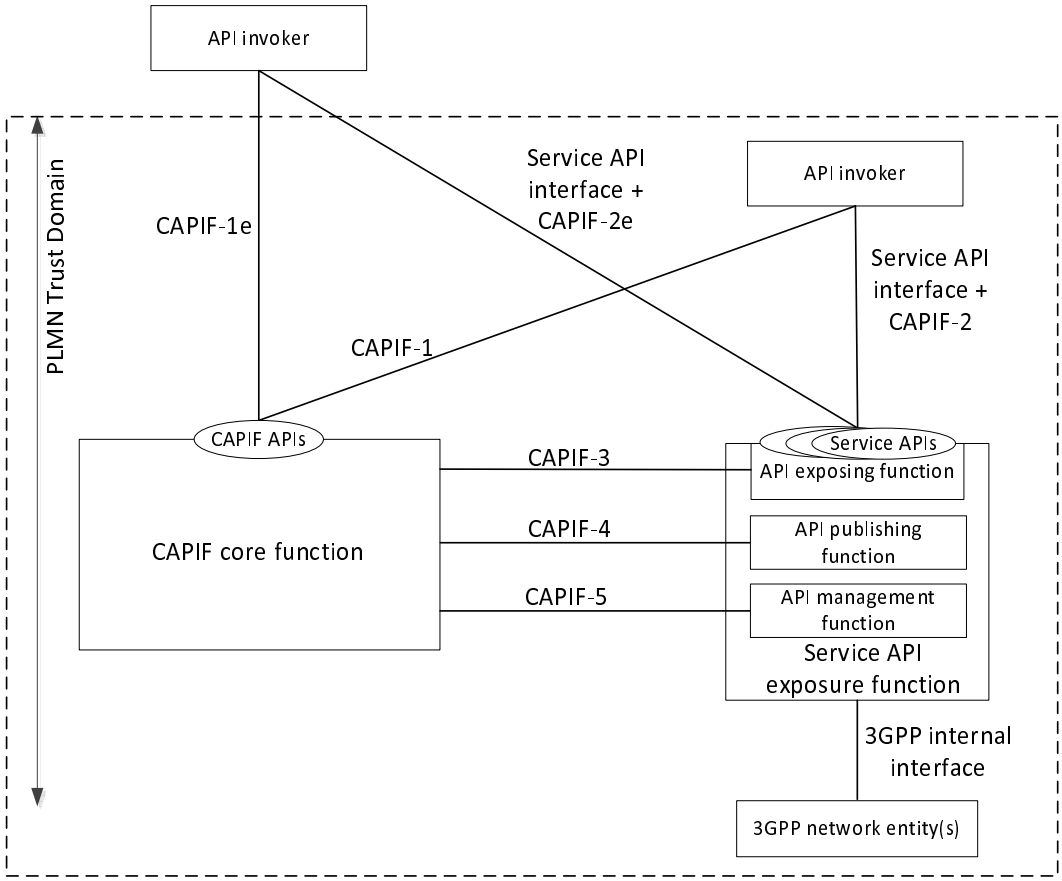

Figure B.0-1: CAPIF utilization by service API provider

The service API aspects of the 3GPP network services and capabilities such as subscriber management, mobility management, transport and other communication services can be exposed for consumption by external 3rd party applications (e.g. API invoker).

Framework aspects typically horizontal in nature caters to common functionality such as onboarding, offboarding, publishing, unpublishing, update service API, discovery, authentication, registration, authorization, logging, charging, monitoring, configuration, topology hiding, that are required to provide service APIs to API invokers. Service APIs can

utilize the functions of the API provider domain (i.e. API exposing function, API publishing function, API management function) and interfaces CAPIF-3, CAPIF-4 and CAPIF-5 as specified in this specification.

The service API exposure function is connected to 3GPP network entity(s) via 3GPP internal interface(s). The API publishing function provides the service API information for publishing to the CAPIF core function.

For consuming service API, the API invoker interacts with the service API exposure function via service API interface and CAPIF-2/2e. While the service API interface is responsible for providing service aspects, CAPIF-2/2e supports service API by providing framework aspects such as authentication of the API invoker, authorization verification for the API invoker upon accessing the service API.

B.1 CAPIF relationship with 3GPP EPS network exposure

B.1.1 General

The table B.1.1-1 shows the relationship between CAPIF and EPS network exposure aspects. The details of SCEF and its role in exposing network capabilities of EPS to 3rd party applications are specified in 3GPP TS 23.682 [2]

Table B.1.1-1: CAPIF relationship with 3GPP EPS network exposure

Aspects	CAPIF	EPS network exposure
Entity providing the APIs to external or 3rd party applications	AEF	SCEF
Entity providing framework related services to the applications (discovery, authentication, authorization, etc)	CAPIF core function	SCEF
Entity representing the external or 3rd party applications	API invoker	SCS/AS
Entity providing framework related services to support the APIs operation and management (publish, policy enforcements, charging)	CAPIF core function	SCEF
Interface/Reference point for exposing network capabilities as APIs	CAPIF-2 and CAPIF-2e (Do not include the service specific aspects)	T8
Interface/Reference point for exposing framework services as APIs to the applications	CAPIF-1 and CAPIF-1e	Not specified. (May be via T8)
Interface/Reference point for framework services to support the APIs operation and management	CAPIF-3, CAPIF-4 and CAPIF-5	Internal to SCEF

B.1.2 Deployment models

B.1.2.1 General

Based on the relationship captured in table B.1.1-1, the following deployment models for CAPIF are possible to enable EPS network exposure.

NOTE: The deployment models captured in subclause 7 are possible for the SCEF deployment compliant with CAPIF. Not all deployment models are illustrated in this subclause.

B.1.2.2 SCEF implements the CAPIF architecture

Figure B.1.2.2-1 illustrates the deployment model where SCEF implements the CAPIF architecture.

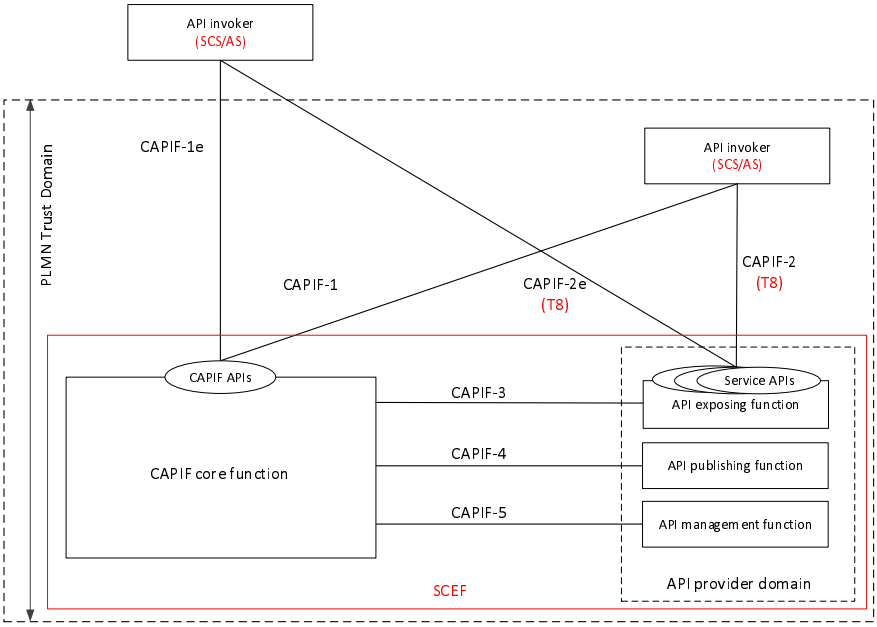

Figure B.1.2.2-1: SCEF implements the CAPIF architecture

The SCEF can implement the functionalities of the CAPIF core function, the API exposing function, the API publishing function and the API management function.

According to the CAPIF architecture, CAPIF-2 and CAPIF-2e consist of framework aspects and service specific aspects. The service specific aspects are out of scope of CAPIF. T8 can implement the service specific aspects of CAPIF-2 and CAPIF-2e, and can provide the service APIs exposed by SCEF (AEF) to the SCS/AS (API invoker).

The SCEF can additionally provide CAPIF-1 and CAPIF-1e (CAPIF APIs) to the SCS/AS (API invokers).

B.1.2.3 SCEF implements the service specific aspect compliant with the CAPIF architecture

Figure B.1.2.3-1 illustrates the deployment model where SCEF implements the service specific aspect compliant with the CAPIF architecture.

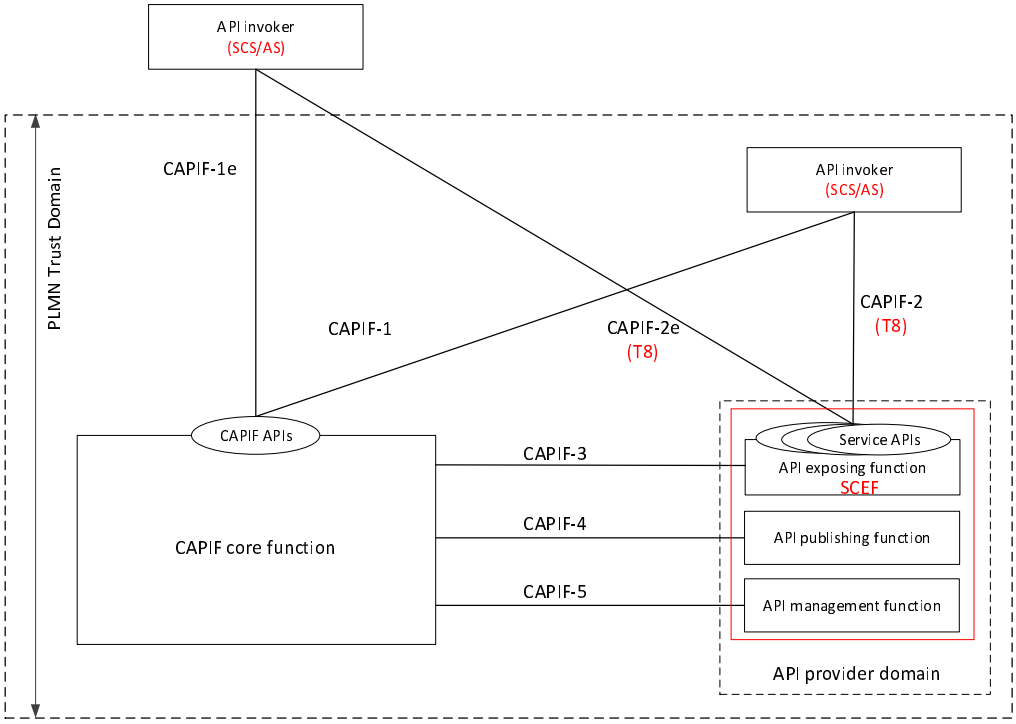

Figure B.1.2.3-1: SCEF implements the service specific aspect compliant with the CAPIF architecture

3GPP EPS can deploy the CAPIF core function along with the SCEF.

The SCEF can implement the functionalities of the API provider domain functions.

According to the CAPIF architecture, CAPIF-2 and CAPIF-2e consist of framework aspects and service specific aspects. The service specific aspects are out of scope of CAPIF. T8 can implement the service specific aspects of CAPIF-2 and CAPIF-2e, and can provide the service APIs exposed by SCEF (AEF) to the SCS/AS (API invoker).

The SCEF can implement the CAPIF-3 reference point/interface to the CAPIF core function.

B.1.2.4 Distributed deployment of the SCEF compliant with the CAPIF architecture

Figure B.1.2.4-1 illustrates the distributed deployment model where the SCEF implements the service specific aspect compliant with the CAPIF architecture.

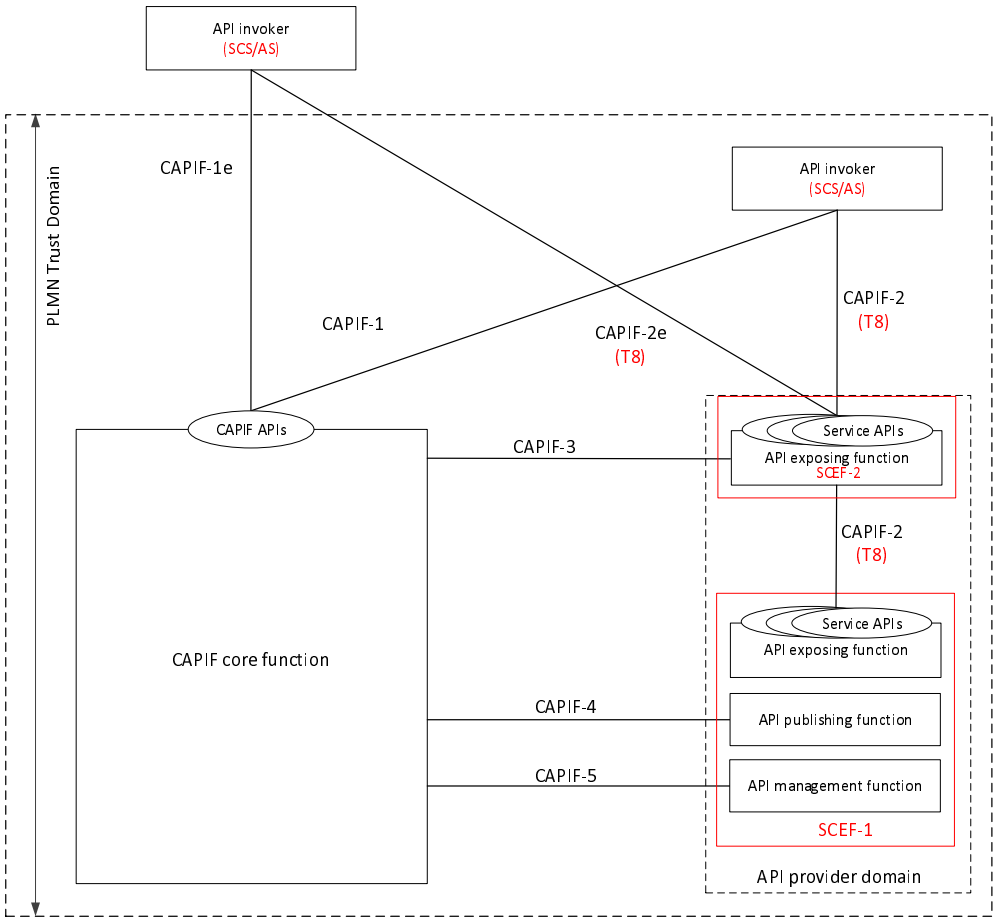

Figure B.1.2.4-1: Distributed deployment of SCEF compliant with the CAPIF architecture

The 3GPP EPS can deploy the CAPIF core function, the SCEF-2 (API exposing function as a gateway) along with the SCEF-1 as illustrated in subclause 7.3.

The SCEF can implement the functionalities of API provider domain functions.

According to the CAPIF architecture, CAPIF-2 or CAPIF-2e consists of framework aspects and service specific aspects. The service specific aspects are out of scope of the CAPIF. T8 can implement the service specific aspects of CAPIF-2 or CAPIF-2e and can provide the service APIs exposed by the SCEF-2 (AEF as a gateway) to the SCS/AS (API invoker).

The SCEF-2 can implement the CAPIF-3 reference point to the CAPIF core function and the SCEF-1 can implement the CAPIF-4 and CAPIF-5 reference points to the CAPIF core function.

Editor's Note: The illustration of this deployment model requires further study.

B.2 CAPIF relationship with 3GPP 5GS network exposure

B.2.1 General

The table B.2.1-1 shows the relationship between CAPIF and 5GS network exposure aspects. The details of NEF and its role in exposing network capabilities of 5GS to 3rd party applications are specified in 3GPP TS 23.501 [3] and the details of NEF service operations are specified in 3GPP TS 23.502 [4].

Table B.2.1-1: CAPIF relationship with 3GPP 5GS network exposure

Aspects	CAPIF	5GS network exposure
Entity providing the APIs to external or 3rd partyapplications	AEF	NEF
Entity providing framework related services to the applications (discovery, authentication, authorization, etc)	CAPIF core function	NEF (Not specified yet)
Entity representing the external or 3rd party applications	API invoker	AF
Entity providing framework related services to support the APIs operation and management (publish, policy enforcements, charging)	CAPIF core function	NEF (Not specified yet)
Interface/Reference point for exposing network capabilities as APIs	CAPIF-2 and CAPIF-2e (Do not include the service specific aspects)	Nnef
Interface/Reference point for exposing framework services as APIs to the applications	CAPIF-1 and CAPIF-1e	Nnef (Not specified yet)
Interface/Reference point for framework services to support the APIs operation and management	CAPIF-3, CAPIF-4 and CAPIF-5	Internal to NEF

B.2.2 Deployment models

B.2.2.1 General

Based on the relationship captured in table B.2.1-1, the following deployment models for CAPIF are possible to enable 5GS network exposure.

NOTE: The deployment models captured in subclause 7 are possible for the NEF deployment compliant with CAPIF. Not all deployment models are illustrated in this subclause.

B.2.2.2 NEF implements the CAPIF architecture

Figure B.2.2.2-1 illustrates the deployment model where the NEF implements the CAPIF architecture.

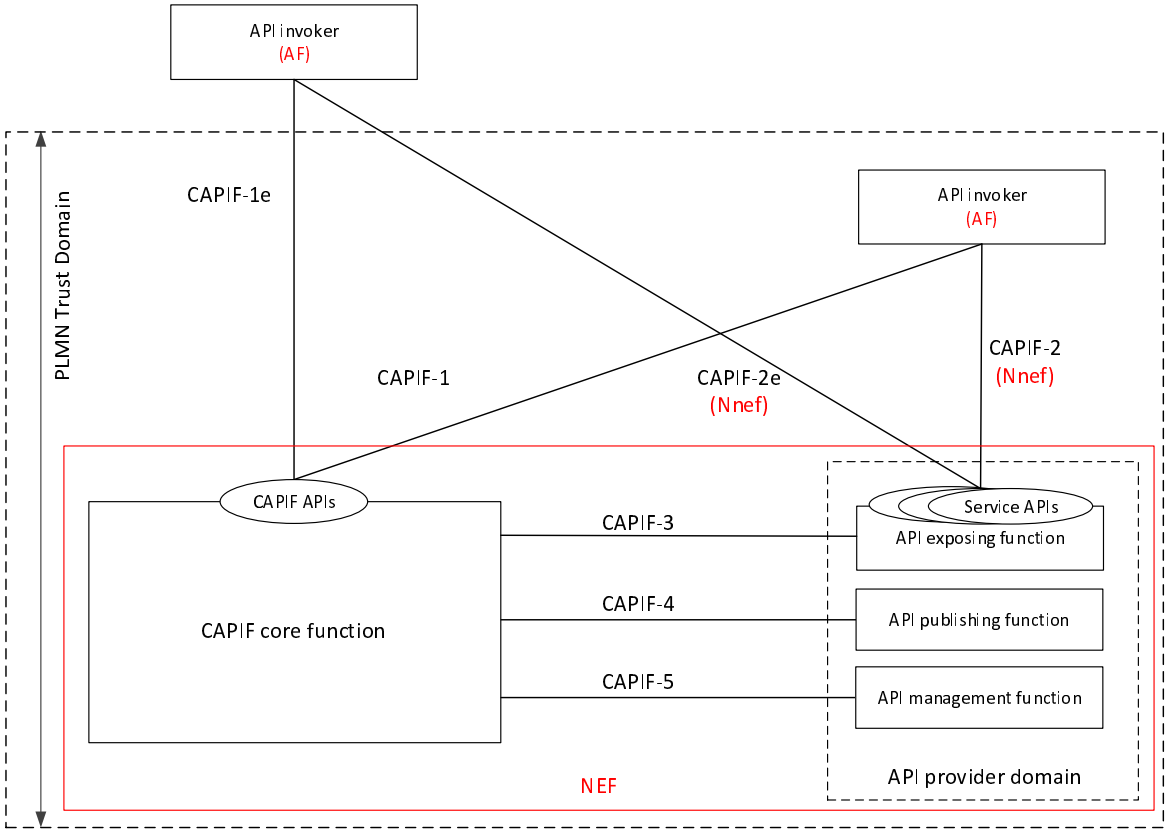

Figure B.2.2.2-1: NEF implements the CAPIF architecture

The NEF can implement the functionalities of the CAPIF core function, the API exposing function, the API publishing function and the API management function.

According to the CAPIF architecture, CAPIF-2 and CAPIF-2e consist of framework aspects and service specific aspects. The service specific aspects are out of scope of CAPIF. Nnef can implement the service specific aspects of CAPIF-2 and CAPIF-2e, and can provide the service APIs exposed by the NEF (AEF) to the AF (API invoker).

The NEF can additionally provide CAPIF-1 and CAPIF-1e (CAPIF APIs) to the AF (API invokers).

B.2.2.3 NEF implements the service specific aspect compliant with the CAPIF architecture

Figure B.2.2.3-1 illustrates the deployment model where the NEF implements the service specific aspect compliant with the CAPIF architecture.

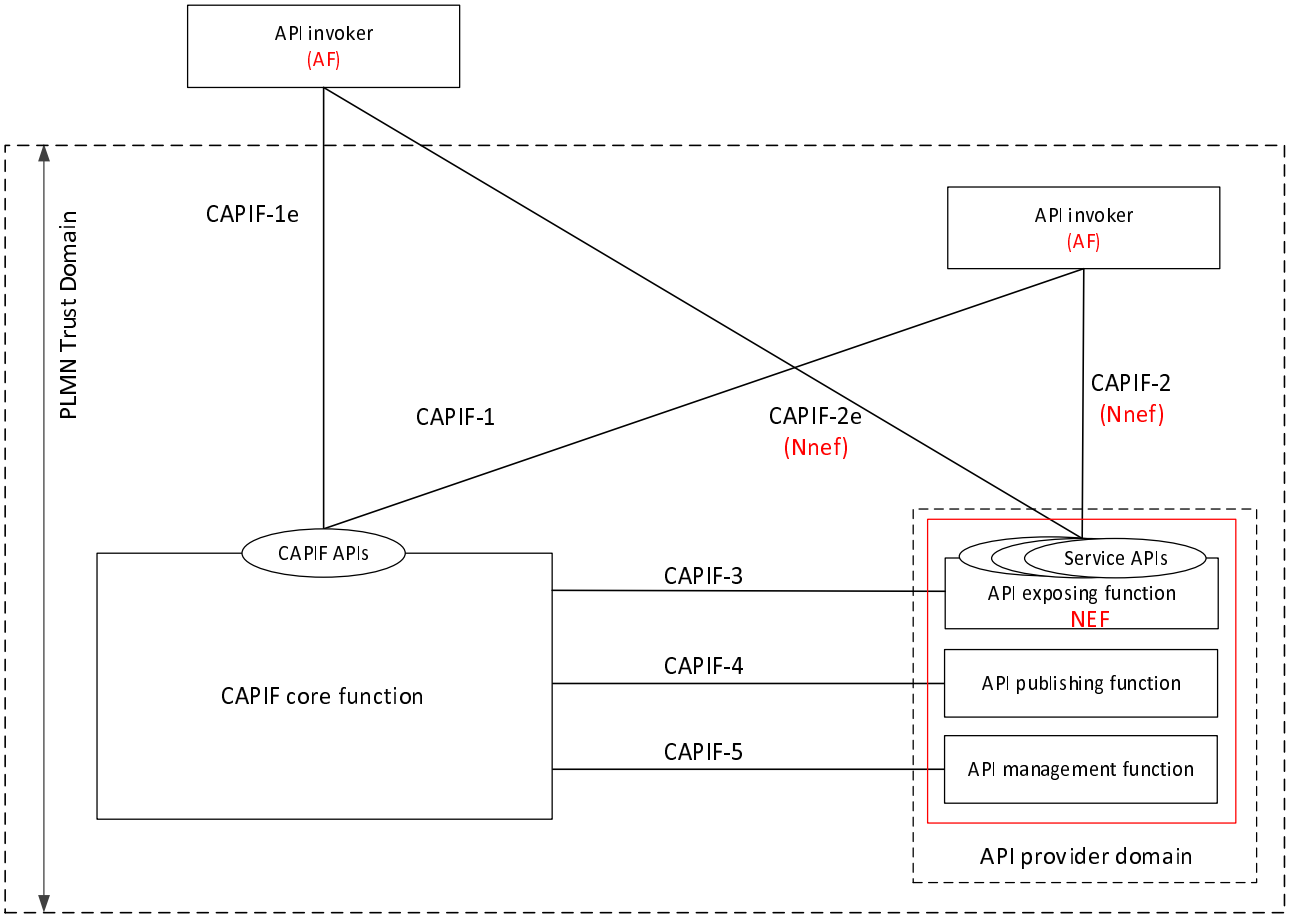

Figure B.2.2.3-1: NEF implements the service specific aspect compliant with the CAPIF architecture

3GPP 5GS can deploy the CAPIF core function along with the NEF.

The NEF can implement the functionalities of the API provider domain functions.

According to the CAPIF architecture, CAPIF-2 and CAPIF-2e consist of framework aspects and service specific aspects. The service specific aspects are out of scope of CAPIF. Nnef can implement the service specific aspects of CAPIF-2 and CAPIF-2e, and can provide the service APIs exposed by NEF (AEF) to the AF (API invoker).

The NEF can implement the CAPIF-3 reference point/interface to the CAPIF core function.

B.2.2.4 Distributed deployment of the NEF compliant with the CAPIF architecture

Figure B.2.2.4-1 illustrates the distributed deployment model where the NEF implements the service specific aspect compliant with the CAPIF architecture.

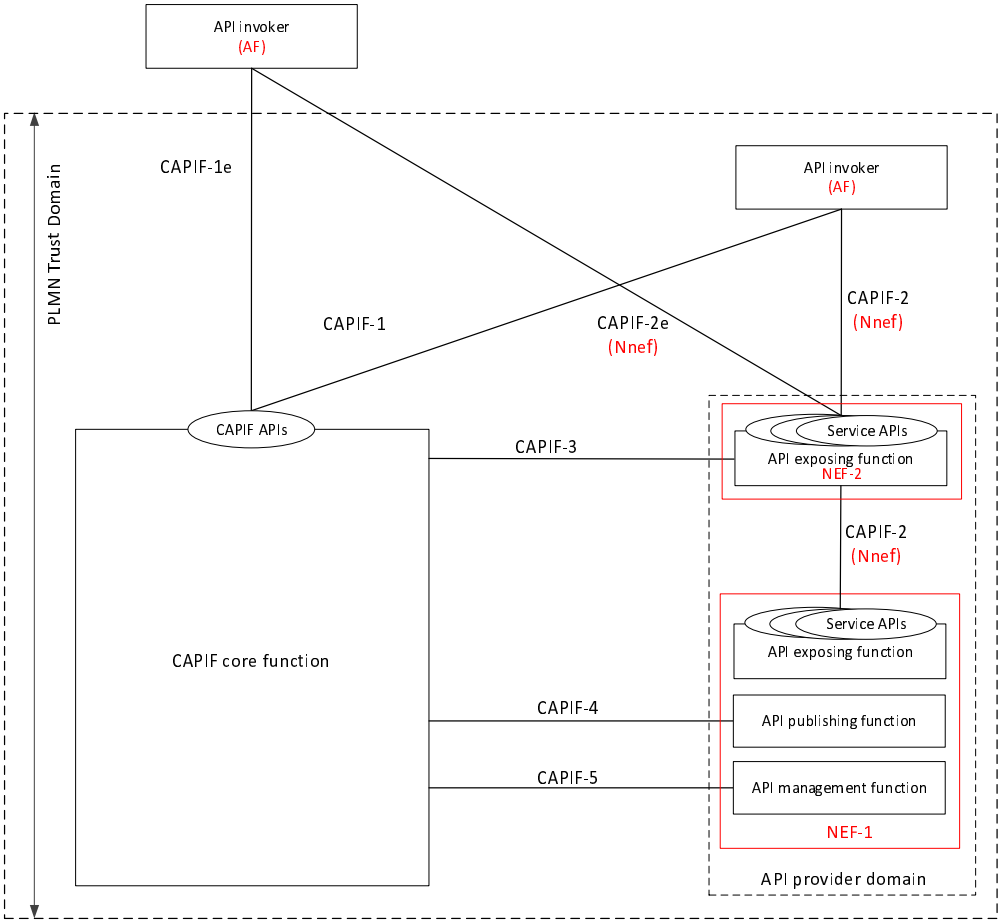

Figure B.2.2.4-1: Distributed deployment of NEF compliant with the CAPIF architecture

The 3GPP 5GS can deploy the CAPIF core function, the NEF-2 (API exposing function as a gateway) along with the NEF-1 as illustrated in subclause 7.3.

The NEF can implement the functionalities of API provider domain functions.

According to the CAPIF architecture, CAPIF-2 or CAPIF-2e consists of framework aspects and service specific aspects. The service specific aspects are out of scope of the CAPIF. Nnef can implement the service specific aspects of CAPIF-2 and CAPIF-2 or CAPIF-2e can provide the service APIs exposed by the NEF-2 (AEF as a gateway) to the AF (API invoker).

The NEF-2 (AEF) can implement the CAPIF-3 reference point to the CAPIF core function and the NEF-1 can implement the CAPIF-4 and CAPIF-5 reference points to the CAPIF core function.

Editor's Note: The illustration of this deployment model requires further study.

B.3 Integrated deployment of 3GPP network exposure systems with the CAPIF

B.3.1 General

According to 3GPP TS 23.682 [2], when the CAPIF is supported, the SCEF supports the API provider domain functions. According to 3GPP TS 23.501 [3], when the CAPIF is supported,the NEF supports the API provider domain functions.

B.3.2 Deployment model

B.3.2.1 General

The SCEF and the NEF may be integrated with a single CAPIF core function to offer their respective service APIs to the API invokers. The following deployment model is possible for integrated deployment of the SCEF and the NEF with the CAPIF core function.

B.3.2.2 Integrated deployment of the SCEF and the NEF with the CAPIF

Figure B.3.2.2-1 illustrates integrated deployment of the SCEF and the NEF with the CAPIF.

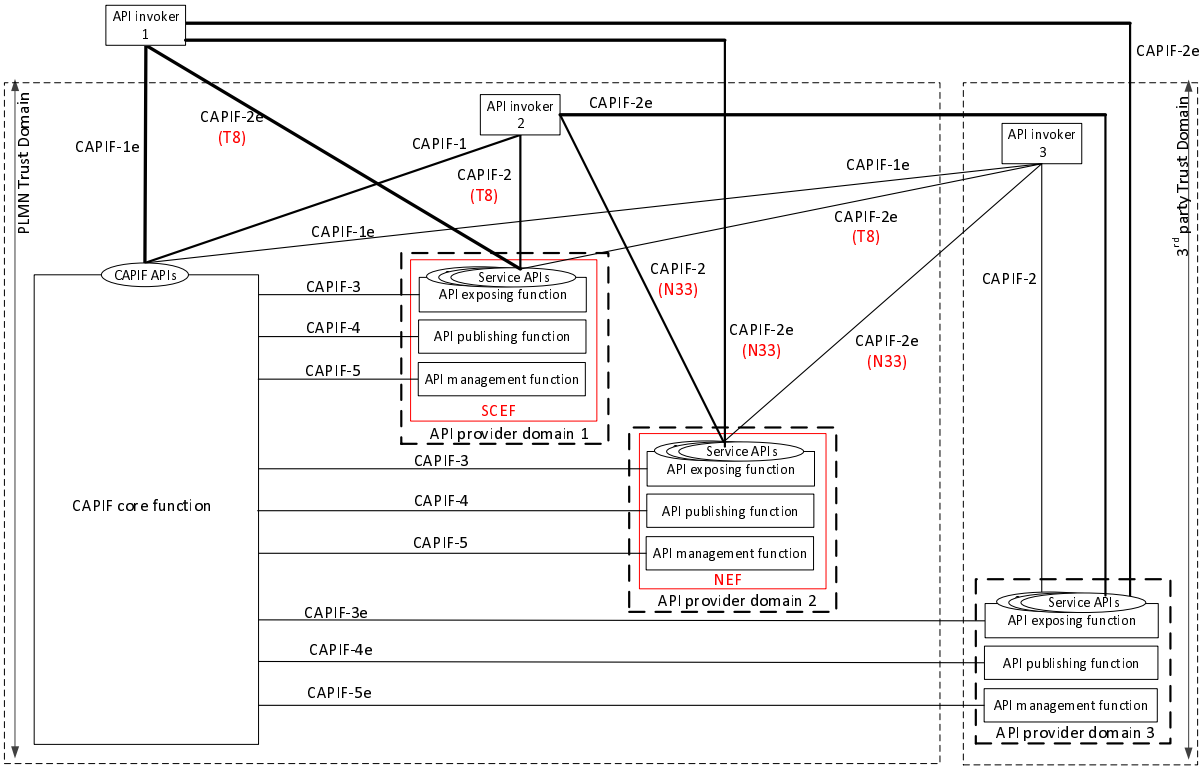

Figure B.3.2.2-1: Integrated deployment of the SCEF and the NEF with the CAPIF

The CAPIF core function, the SCEF and the NEF are deployed in the PLMN trust domain, where the CAPIF core function takes the role of a unified gateway and provides services to different API invokers. The API invokers obtains the T8 and N33 service API information and the corresponding entry point details from the CAPIF core function via CAPIF-1 or CAPIF-1e reference points.

The API invokers can interact independently with the SCEF, the NEF and the 3rd party API exposing functions via CAPIF-2 or CAPIF-2e reference points. In this case, T8 and N33 can be reused to implement the service specific aspects of CAPIF-2 or CAPIF-2e reference points for the corresponding service API interactions of the SCEF and the NEF respectively.

The SCEF and the NEF applies any service API access policy control to the interactions between the API invokers and the T8 and N33 service APIs respectively by communicating with the same CAPIF core function via the CAPIF-3 reference point.

Annex C (informative): CAPIF role in charging

C.1 General

This annex provides the information about the role of CAPIF in charging service API invocations. The common architecture for charging is illustrated in clause 4 of 3GPP TS 32.240 [6]. There are two charging mechanisms - offline charging and online charging. The role of CAPIF in both these charging mechanims is illustrated for informational purpose in this subclause.

The API invocations are subjected to charging (online, offline) as illustrated in figure C.1-1.

NOTE: As there are no impacts on CAPIF's role in charging due to deployment of 3rd party trust domain, it is not illustrated in the figures.

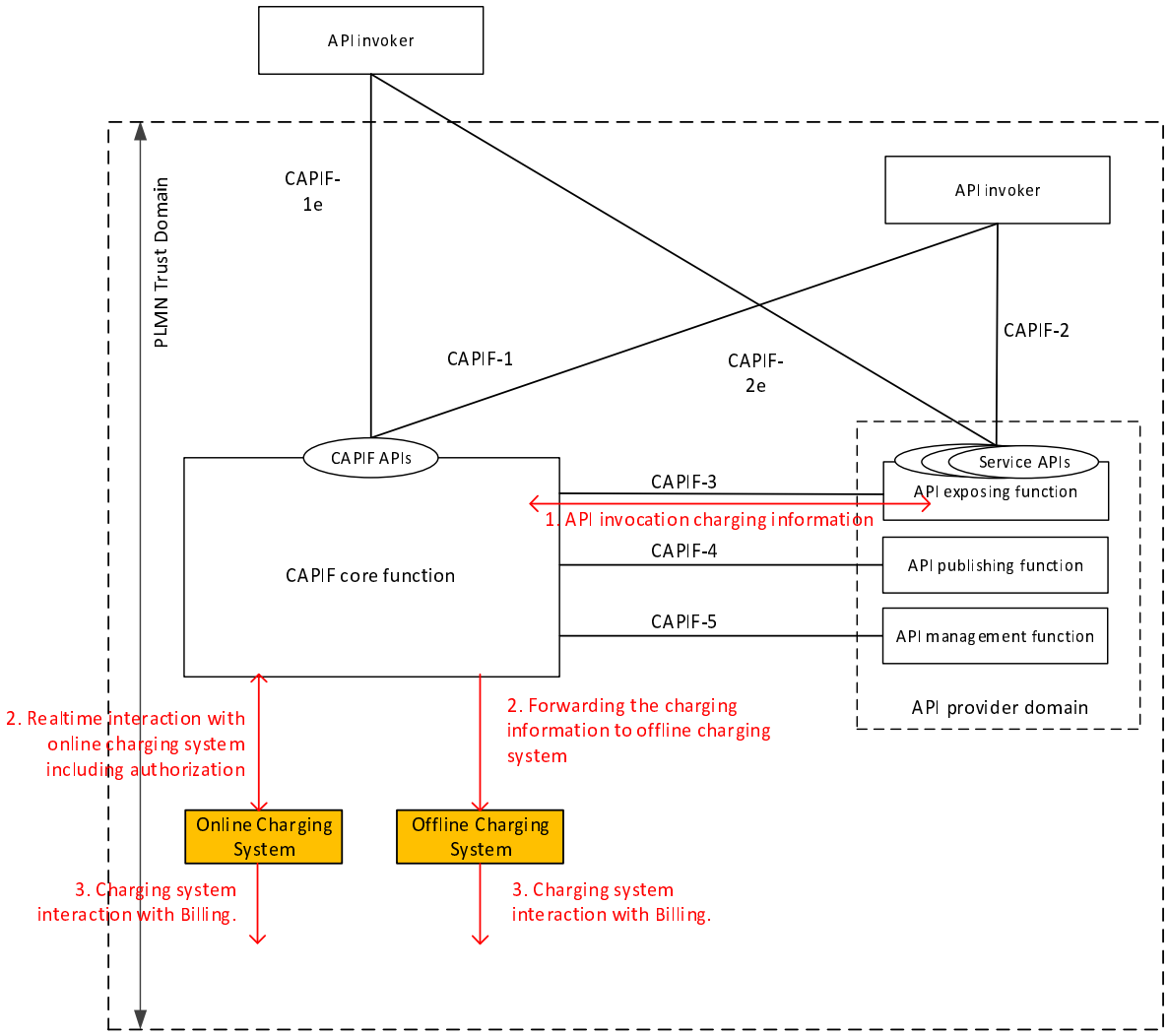

Figure C.1-1: CAPIF role in charging

C.2 CAPIF role in online charging

The API invocations are subjected to online charging as illustrated in figure C.1-1.

The API exposing function provides the API invocation charging information to the CAPIF core function. The CAPIF core function further interacts with an online charging system in real-time by providing the charging information and further the CAPIF core function receives the authorization corresponding to the charging information.

NOTE: The online charging architecture for CAPIF including specification of online charging system entities and reference points is under the responsibility of SA5.

C.3 CAPIF role in offline charging

The API invocations are subjected to offline charging as illustrated in figure C.1-1.

The API exposing function provides the API invocation charging information to the CAPIF core function. The CAPIF core function provides the charging information to the offline charging system. The offline charging system generates the CDRs for the API invocation and further transfers the CDR files to the billing domain.

NOTE: The offline charging architecture for CAPIF including specification of offline charging system entities and reference points is under the responsibility of SA5.

Annex D (informative): CAPIF relationship with external API frameworks

This annex provides the relationship of CAPIF with the OMA Network APIs and the ETSI MEC API framework. The relationship of CAPIF with these external API frameworks is illustrated in the table D-1. "Yes" means that the external API framework supports the CAPIF functionality, "No" means that the API framework does not support the CAPIF functionality, and "Partial" means that it provides a mechanism that partially supports the CAPIF functionality.

Table D-1: CAPIF relationship with external API frameworks

CAPIF functionalities	OMA Network APIs		ETSI MEC API framework	
	Supported	**Reference**	**Supported**	**Reference**
Publish and discover service API information	Partial (see NOTE)	OMA-TS-NGSI_Registration_and_Discovery [11]	Yes	ETSI GS MEC 011 [7]
Topology hiding of the service	Yes	Individual API exposing function	Yes	Individual API exposing function
API invoker authentication to access service APIs	Partial	OMA-ER_Autho4API [9]	Partial	ETSI GS MEC 009 [8]
API invoker authorization to access service APIs	Partial	OMA-ER_Autho4API [9]	Partial	ETSI GS MEC 009 [8]
Charging on invocation of service APIs	No		No	
Lifecycle management of service APIs	No		No	
Monitoring service API invocations	No		No	
Logging API invoker onboarding and service API invocations	No		No	
Auditing service API invocations	No		No	
Onboarding API invoker to CAPIF	No		No	
CAPIF authentication of API invokers	No		No	
Service API access control	Partial	OMA-ER_Autho4API [9]	Partial	ETSI GS MEC 009 [8]
Secure API communication	Yes	OMA-ER_Autho4API [9]	Yes	ETSI GS MEC 009 [8]
Policy configuration	No		No	
API protocol stack model	Partial	for REST: OMA-TS_REST_NetAPI_Common [10]	Partial	for REST: ETSI GS MEC 009 [8]
API security protocol	Partial	OMA-ER_Autho4API [9]	Partial	ETSI GS MEC 009 [8]
CAPIF support for service APIs from multiple providers	No		No	
NOTE: OMA-TS-NGSI_Registration_and_Discovery [11] is only applicable to a specific type of web services (OWSER using UDDI and WSDL).				

Annex E (normative): Configuration data for CAPIF

The configuration data is stored in the CAPIF core function and provided by the CAPIF administrator.

The configuration data for CAPIF is specified in table E-1.

Table E-1: Configuration data for CAPIF

Reference	Parameter description
Subclause 4.2.2	List of published service API discovery restrictions
	> Service API identification
	> API invoker identity information
Subclause 4.7.2	List of service API log storage durations
	> Service API identification
	> Service API log storage duration (in hours) (see NOTE)
Subclause 4.7.4	List of API invoker interactions log storage durations
	> Service API identification
	API invoker interactions log storage duration (in hours) (see NOTE)
Subclause 4.10	List of access control policy per API invoker
	> Volume limit on service API invocations (total number of invocations allowed)
	> Time limit on service API invocations (The time range of the day during which the service API invocations are allowed)
	> Rate limit on service API invocations (allowed service API invocations per second)
	> Service API identification
	> API invoker identity information
NOTE: If no value is set for the duration, the duration is assumed to be unlimited.	

Annex F (informative): Change history

Change history							
Date	**Meeting**	**TDoc**	**CR**	**Rev**	**Cat**	**Subject/Comment**	**New version**
2017-10	SA6#19	S6-171274				TS skeleton	0.0.0
2017-10	SA6#19					Implementation of the following p-CRs approved by SA6: S6-171444; S6-171343; S6-171445; S6-171446; S6-171466; S6-171448; S6-171348; S6-171449; S6-171359; S6-171467; S6-171451; S6-171452; S6-171362; S6-171463; S6-171356; S6-171355; S6-171453; S6-171454; S6-171455; S6-171464; S6-171468; S6-171350; S6-171349; S6-171407.	0.1.0
2017-12	SA6#20					Implementation of the following p-CRs approved by SA6: S6-171630; S6-171631; S6-171633; S6-171648; S6-171650; S6-171658; S6-171659; S6-171692; S6-171693; S6-171694; S6-171695; S6-171698; S6-171699; S6-171700; S6-171702; S6-171704; S6-171705; S6-171706; S6-171711; S6-171712; S6-171713; S6-171819; S6-171820; S6-171821; S6-171822; S6-171823; S6-171848; S6-171855; S6-171865; S6-171876.	0.2.0
2017-12	SA#78	SP-170901				Submitted to SA#78 for approval	1.0.0
2018-01	SA#78	SP-170901				MCC Editorial update for publication after TSG SA approval (SA#78)	15.0.0
2018-04	SA#79	SP-180156	0001	1	F	Use of specific ETSI and OMA references	15.1.0
2018-04	SA#79	SP-180156	0002		F	Corrections for CAPIF-1e and CAPIF-2e	15.1.0
2018-04	SA#79	SP-180156	0003		F	Miscellaneous corrections to procedures and information flows	15.1.0
2018-04	SA#79	SP-180156	0004	1	F	Addition of offboarding to functional entities and reference points description	15.1.0
2018-04	SA#79	SP-180156	0005	1	D	Editorial corrections	15.1.0
2018-04	SA#79	SP-180156	0006	2	B	Solution to EN on revoking authorization based on access control	15.1.0
2018-04	SA#79	SP-180156	0007	3	F	Configuration items for CAPIF	15.1.0
2018-04	SA#79	SP-180156	0008	3	F	Update to CAPIF relationship with 3GPP EPS and 5GS	15.1.0
2018-04	SA#79	SP-180156	0009	1	F	Solution to EN on policy synchronization	15.1.0
2018-04	SA#79	SP-180156	0010	2	F	CAPIF utilization by service APIs	15.1.0
2018-04	SA#79	SP-180156	0011	1	F	Proposal for definition for PLMN trust domain	15.1.0
2018-06	SA#80	SP-180374	0013	1	F	Correction for the details of service API information	15.2.0
2018-06	SA#80	SP-180374	0014	1	F	Correction for usage of service API identification information	15.2.0
2018-06	SA#80	SP-180374	0019	2	D	Editorial correction of TS 23.222 (CAPIF stage2)	15.2.0
2018-06	SA#80	SP-180375	0012	2	B	Architecture functional model to support multiple API providers	16.0.0
2018-06	SA#80	SP-180375	0015	1	B	Service API publish and discovery requirements for 3rd party API providers	16.0.0
2018-06	SA#80	SP-180375	0016	1	B	Charging requirements for 3rd party API providers	16.0.0
2018-06	SA#80	SP-180375	0017	1	B	OAM requirements for 3rd party API providers	16.0.0
2018-06	SA#80	SP-180375	0018	2	B	CAPIF interconnection requirements	16.0.0
2018-06	SA#80	SP-180375	0020	2	F	Updating representation of deployment models	16.0.0
2018-09	SA#81	SP-180675	0021	2	B	Integrated CAPIF with 3GPP EPS and 5GS network exposure	16.1.0
2018-09	SA#81	SP-180675	0022	1	C	Enhancement to the functional model deployments	16.1.0
2018-09	SA#81	SP-180675	0023	2	B	Enhancement to reference points for eCAPIF	16.1.0
2018-09	SA#81	SP-180674	0029	1	A	Update API naming convention	16.1.0
2018-09	SA#81	SP-180674	0030	2	A	Alignment of APIs	16.1.0
2018-09	SA#81	SP-180674	0031	1	A	Alignment to SA3 CAPIF TS	16.1.0
2018-09	SA#81	SP-180674	0032	1		Alignment to SA3 authentication procedure	16.1.0
2018-09	SA#81	SP-180675	0033	3	B	Functional architecture for CAPIF interconnection	16.1.0
2018-12	SA#82	SP-181176	0034	3	B	Topology hiding enhancement	16.2.0
2018-12	SA#82	SP-181176	0035	2	B	API publish and API discover for CAPIF interconnection	16.2.0
2018-12	SA#82	SP-181176	0036	1	C	Architectural requirements for identities	16.2.0
2018-12	SA#82	SP-181176	0038	2	B	Architectural requirements for provider domain entities interaction	16.2.0
2018-12	SA#82	SP-181176	0039	2	B	Update API invoker API list	16.2.0
2018-12	SA#82	SP-181175	0043	2	A	API invoker's onboarding response rel16	16.2.0
2019-03	SA#83	SP-190072	0044	2	F	Update procedures with topology hidding	16.3.0
2019-03	SA#83	SP-190072	0045	2	B	API sharing for CCF interconnection	16.3.0
2019-03	SA#83	SP-190072	0046	2	B	API invocation request routing with topology hiding	16.3.0
2019-03	SA#83	SP-190072	0048	1	C	Interactions between API exposing functions	16.3.0
2019-03	SA#83	SP-190072	0049	1	B	Service API discovery involving multiple CCFs	16.3.0
2019-03	SA#83	SP-190072	0050	2	B	Multiple CCFs deployment in a PLMN trust domain	16.3.0
2019-03	SA#83	SP-190072	0051	2	B	Service API discover for CAPIF interconnection	16.3.0
2019-03	SA#83	SP-190072	0052	1	B	Architectural requirements for registration of API provider domain functions	16.3.0
2019-03	SA#83	SP-190072	0053	2	B	Procedures for registration of API provider domain functions	16.3.0
2019-03	SA#83	SP-190072	0054	1	B	Updates to AEF procedures for 3rd party trust domain	16.3.0
2019-03	SA#83	SP-190072	0055	1	B	Updates to APF procedures for 3rd party trust domain	16.3.0
2019-03	SA#83	SP-190072	0056	1	B	Updates to AMF procedures for 3rd party trust domain	16.3.0
2019-03	SA#83	SP-190072	0057	-	B	Updates to CAPIF events procedures for 3rd party trust domain	16.3.0
2019-06	SA#84	SP-190483	0058	1	F	Clarification to routing rule of service API invocation	16.4.0
2019-06	SA#84	SP-190483	0059	3	F	Functional model update with reference points	16.4.0

2019-06	SA#84	SP-190483	0060	2	B	Update to service API publish for CAPIF interconnection	16.4.0
2019-06	SA#84	SP-190483	0061	2	B	Serving area and domain of service API for CAPIF interconnection	16.4.0
2019-06	SA#84	SP-190483	0062	1	B	3rd party trust domain with network exposure and charging aspects of 3GPP systems	16.4.0
2019-06	SA#84	SP-190483	0063	1	B	Interface based representation of CAPIF architecture	16.4.0
2019-09	SA#85	SP-190828	0064	1	F	Clarification and alignment with publish request information flows	16.5.0
2019-12	SA#86	SP-191107	0065		F	Correction on usage of service API information in access control message	16.6.0
2020-03	SA#87-E	SP-200112	0066	1	F	Shared CAPIF provider domain info in interconnection	16.7.0
2020-03	SA#87-E	SP-200116	0067	2	B	Serving area information for service APIs to support edge applications	17.0.0
2020-07	SA#88-E	SP-200337	0069		A	Add consumer for discover and publish service APIs	17.1.0
2020-07	SA#88-E	SP-200337	0071		A	Add obtaining routing info service API	17.1.0
2020-07	SA#88-E	SP-200337	0073	1	A	Correct API topology hiding	17.1.0
2020-07	SA#88-E	SP-200337	0075	1	A	Correction for CAPIF interconnection IEs	17.1.0
2020-09	SA#89-E	SP-200840	0077	3	F	Correction for API routing information	17.2.0
2020-12	SA#90-E	SP-200997	0078	2	B	Support AEF location and API invoker interface for edge application	17.3.0
2021-04	SA#91-E	SP-210183	0079	4	F	Clarification of Service-based interfaces interaction within CAPIF	17.4.0
2021-06	SA#92-E	SP-210482	0082		A	API provider management API	17.5.0
2022-06	SA#96	SP-220471	0084	1	A	Corrections to API invoker onboarding/offboarding in TS 23.222	17.6.0
2022-09	SA#97	SP-220918	0089		A	Corrections to API invoker onboarding/offboarding in TS 23.222	17.7.0
2022-12	SA#98-e	SP-221250	0090	1	B	Additional CAPIF architectural requirements for SNA	18.0.0
2022-12	SA#98-e	SP-221250	0091	2	B	CAPIF business relationship updates for SNA	18.0.0
2022-12	SA#98-e	SP-221250	0092	2	B	CAPIF functional model updates for SNA	18.0.0
2022-12	SA#98-e	SP-221250	0093	2	B	API invoker obtaining authorization from resource owner	18.0.0
2022-12	SA#98-e	SP-221250	0094	1	B	Discover a proper AEF with owner information	18.0.0
2022-12	SA#98-e	SP-221250	0095	2	B	Reducing resource owner consent inquiry in a nested API invocation	18.0.0
2022-12	SA#98-e	SP-221239	0096	2	B	CAPIF extensibility as requested by ETSI ISG MEC	18.0.0
2023-03	SA#99	SP-230295	0098		B	Discover proper AEF in interconnection	18.1.0
2023-03	SA#99	SP-230296	0099	1	B	Solve CAPIF extensibility EN	18.1.0
2023-03	SA#99	SP-230295	0100	1	B	API invoker clarification	18.1.0
2023-03	SA#99	SP-230295	0101	1	D	Modify a terminology for SNA	18.1.0
2023-03	SA#99	SP-230286	0102	2	B	New IE(Service KPI) in Service API publish request	18.1.0
2023-03	SA#99	SP-230295	0103	2	B	Discover proper AEF with IP information	18.1.0
2023-03	SA#99	SP-230292	0104		B	Support onboarding expiration	18.1.0
2023-03	SA#99	SP-230296	0105	2	F	Resolving editor's notes about TS reference	18.1.0
2023-03	SA#99	SP-230295	0106	2	B	Adding descriptions of new functional entities and reference points	18.1.0
2023-06	SA#100	SP-230714	0109	2	B	Support CAPIF in SNPN	18.2.0
2023-06	SA#100	SP-230712	0111	4	B	Service API status monitoring	18.2.0
2023-06	SA#100	SP-230713	0112		B	Clarification that RNAA is for both 4G and 5G	18.2.0
2023-06	SA#100	SP-230713	0113	2	B	SNAAPP alignment with SA3	18.2.0
2023-06	SA#100	SP-230713	0114	2	B	Overview of CAPIF operations for RNAA scenarios	18.2.0
2023-06	SA#100	SP-230714	0115	3	B	CAPIF add service procedure for update of subscriptions	18.2.0
2023-06	SA#100	SP-230714	0116	5	B	Alignment among CAPIF provider (trust) domains	18.2.0
2023-12	SA#102	SP-231547	0122	2	A	Editorial corrections	18.3.0
2023-12	SA#102	SP-231569	0125	1	F	Solve EN related to SA3	18.3.0
2023-12	SA#102	SP-231569	0126	2	F	Add response to RNAA procedural flows and correct cross-references	18.3.0
2023-12	SA#102	SP-231569	0128	2	F	Clarify how to monitor service API status when the APF is unable to update service API status	18.3.0
2023-12	SA#102	SP-231569	0129	2	F	Add CAPIF words to Abbreviations	18.3.0
2023-12	SA#102	SP-231569	0133	6	F	CAPIF Architecture alignment with SA3 RNAA aspects	18.3.0
2023-12	SA#102	SP-231546	0136		A	API Description Correction	18.3.0
2023-12	SA#102	SP-231569	0137	1	F	API invoker authorization corrections	18.3.0
2023-12	SA#102	SP-231545	0141		A	Security API corrections	18.3.0
2023-12	SA#102	SP-231569	0147	2	F	Editorial corrections regarding RNAA	18.3.0
2023-12	SA#102	SP-231569	0149	2	F	Corrections for CAPIF revocation of API Invoker's authorization based on RNAA	18.3.0
2023-12	SA#102	SP-231569	0150	1	F	Corrections for CAPIF deployment models supporting RNAA	18.3.0
2024-03	SA#103	SP-240310	0153	2	F	Consistent use of term "resource owner"	18.4.0
2024-06	SA#104	SP-240755	0161	1	F	Corrections to Deregister_API_Provider operation	18.5.0
2024-06	SA#104	SP-240755	0166	2	F	Correction for Discover service APIs	18.5.0
2024-06	SA#104	SP-240764	0168	1	F	Add missing function to resource owner function	18.5.0
2024-06	SA#104	SP-240756	0173	2	F	Alignment of "API type" with "API category" terminology	18.5.0
2024-06	SA#104	SP-240755	0179	2	F	Correction for Service API discovery involving multiple CCFs	18.5.0
2024-06	SA#104	SP-240764	0184	3	F	Correction to clause 6.2.3	18.5.0
2024-06	SA#104	SP-240764	0186		F	Reducing authorization information inquiry in a nested API invocation	18.5.0

History

Document history		
V18.4.0	April 2024	Publication
V18.5.0	July 2024	Publication