ETSI TS 129 222 V15.1.0 (2018-10)

5G;
Common API Framework for 3GPP Northbound APIs
(3GPP TS 29.222 version 15.1.0 Release 15)

Reference
RTS/TSGC-0329222vf10

Keywords
5G

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 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is 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 IPR Policy, no investigation, 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.

Foreword

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, UMTS identities or GSM identities. These should be interpreted as being references to the corresponding ETSI deliverables.

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

Modal verbs terminology

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

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

Contents

Foreword

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

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

Version x.y.z

where:

- x the first digit:
  - 1 presented to TSG for information;
  - 2 presented to TSG for approval;
  - 3 or greater indicates TSG approved document under change control.
- y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.
- z the third digit is incremented when editorial only changes have been incorporated in the document.

1 Scope

The present specification describes the protocol for the Common API Framework (CAPIF) for 3GPP Northbound APIs. The CAPIF and the related stage 2 architecture and functional requirements are defined in 3GPP TS 23.222 [2].

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 23.222: "Functional architecture and information flows to support Common API Framework for 3GPP Northbound APIs; Stage 2".

[3] Open API Initiative, "OpenAPI 3.0.0 Specification", https://github.com/OAI/OpenAPI-Specification/blob/master/versions/3.0.0.md.

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

[5] IETF RFC 7231: "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content".

[6] IETF RFC 7232: "Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests".

[7] IETF RFC 7233: "Hypertext Transfer Protocol (HTTP/1.1): Range Requests".

[8] IETF RFC 7234: "Hypertext Transfer Protocol (HTTP/1.1): Caching".

[9] IETF RFC 7235: "Hypertext Transfer Protocol (HTTP/1.1): Authentication".

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

[11] IETF RFC 5246, "The Transport Layer Security (TLS) Protocol Version 1.2".

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

[13] IETF RFC 6455: "The Websocket Protocol".

[14] 3GPP TS 29.122: "T8 reference point for northbound Application Programming Interfaces (APIs)".

[15] 3GPP TS 29.522: "5G System; Network Exposure Function Northbound APIs; Stage 3".

[16] 3GPP TS 33.122: "Security Aspects of Common API Framework for 3GPP Northbound APIs"

[17] IETF RFC 2617: "HTTP Authentication: Basic and Digest Access Authentication".

[18] 3GPP TS 29.501: "5G System; Principles and Guidelines for Services Definition; Stage 3".

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 registry: API registry is a registry maintained by the CAPIF core function to store information about the service APIs based on the data models defined in this specification. The structure of the API registry is out of scope of this specification.

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

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

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.

Subscriber: A functional entity that subscribes to another functional entity for notifications.

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].

AEF	API Exposing Function
AMF	API Management Function
APF	API Publishing Function
AS	Application Server
CAPIF	Common API Framework
CCF	CAPIF Core Function
JSON	JavaScript Object Notation
REST	Representational State Transfer
SCEF	Service Capability Exposure Function
SCS	Service Capability Server

4 Overview

4.1 Introduction

In 3GPP, there are multiple northbound API-related specifications. To avoid duplication and inconsistency of approaches between different API specifications and to specify common services (e.g. authorization), 3GPP has considered in 3GPP TS 23.222 [2] the development of a common API framework (CAPIF) that includes common aspects applicable to any northbound service APIs.

The present document specifies the APIs needed to support CAPIF.

4.2 Service Architecture

3GPP TS 23.222, clause 6 [2] specifies the functional entities and domains of the functional model, which is depicted in Figure 4.2-1, in detail.

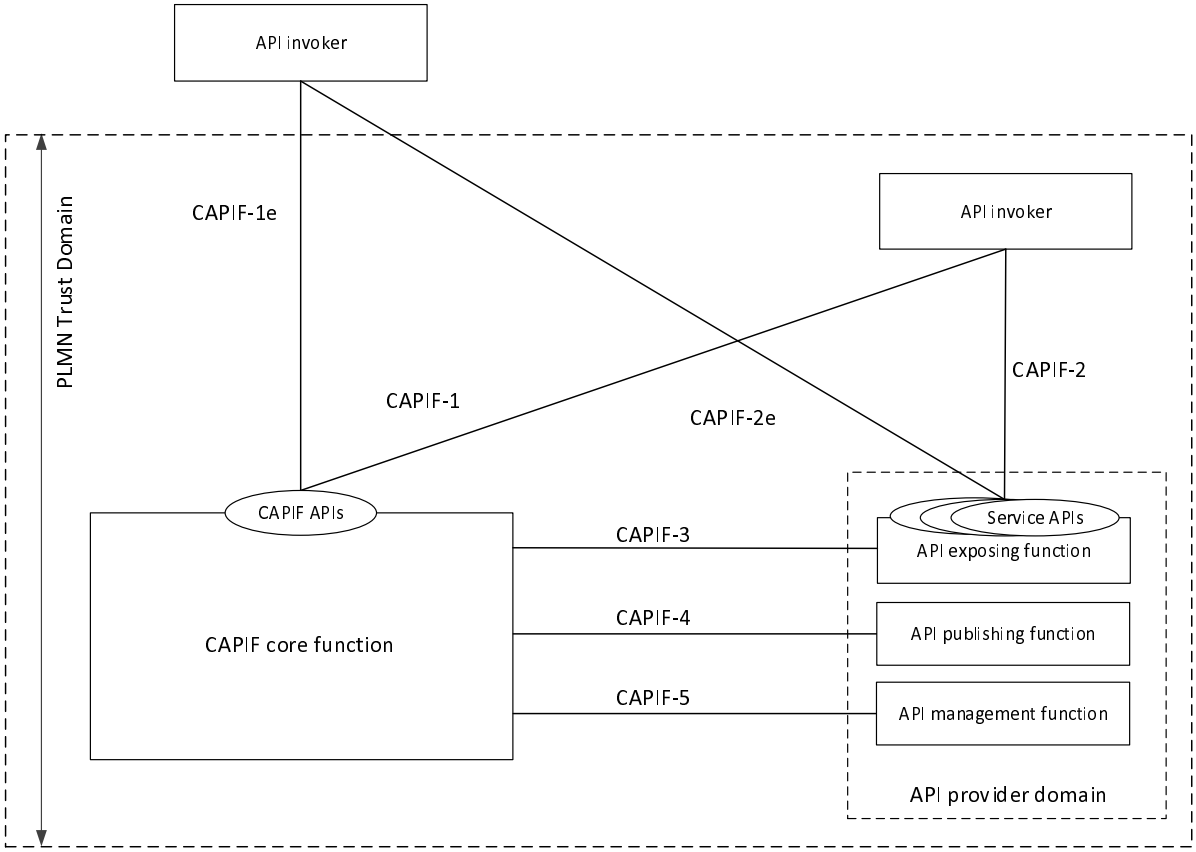

Figure 4.2-1: CAPIF Functional Model

CAPIF-1 and CAPIF-1e reference points connect an API invoker inside the PLMN Trust Domain and an API invoker outside the PLMN Trust Domain respectively, with the CAPIF core function.

CAPIF-2 and CAPIF-2e reference points connect an API invoker inside the PLMN Trust Domain and an API invoker outside the PLMN Trust Domain respectively, with the API exposing function.

CAPIF-3 reference point connects an API exposing function inside the PLMN Trust Domain with the CAPIF core function.

CAPIF-4 reference point connects an API publishing function inside the PLMN Trust Domain with the CAPIF core function.

CAPIF-5 reference point connects an API management function inside the PLMN Trust Domain with the CAPIF core function.

NOTE: The API exposing function, API publishing function and API management function are part the API provider domain which can be implemented by the Service Capability Exposure Function (SCEF) and/or the Network Exposure Function (NEF).

4.3 Functional Entities

4.3.1 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 supports several capabilities such as supporting

- the authentication and obtaining authorization and discovering using CAPIF-1/CAPIF-1e reference point as defined in 3GPP TS 23.222 [2]; and

- invoking the Service APIs using CAPIF-2/CAPIF-2e referenced point as defined in 3GPP TS 23.222 [2], e.g. the T8 interface as defined in 3GPP TS 29.122 [14] or the NEF Northbound interface as defined in 3GPP TS 29.522 [15].

4.3.2 CAPIF core function

The CAPIF core function (CCF) supports the following capabilities over CAPIF-1/CAPIF-1e reference point as defined in 3GPP TS 23.222 [2]:

- authenticating the API invoker;
- providing the authorization information; and
- service API discovery.

The CAPIF core function supports the following capabilities over CAPIF-3 reference point as defined in 3GPP TS 23.222 [2]:

- providing the service API access policy;
- providing the authentication and authorization information of API invoker for validation;
- logging of service API invocations and
- charging of service API invocations.

The CAPIF core function supports the following capabilities over CAPIF-4 reference point as defined in 3GPP TS 23.222 [2]:

- publishing and storing the service APIs information.

The CAPIF core function supports the following capabilities over CAPIF-5 reference point as defined in 3GPP TS 23.222 [2]:

- providing the service API invocation log for auditing;
- providing monitoring information the status of service APIs and
- storing configurations of the API provider policies.

4.3.3 API exposing function

The API exposing function (AEF) is the provider of the Service APIs and is also the service communication entry point of the Service API to the API invokers using CAPIF-2/CAPIF-2e reference point as defined in 3GPP TS 23.222 [2]. The API exposing function consists of capabilities such as authenticating the API invoker, validating the authorization provided by the CAPIF core function and logging the Service API invocations at the CAPIF core function using CAPIF-3 reference point as defined in 3GPP TS 23.222 [2].

According to the distributed deployment scenarios specified in 3GPP TS 23.222 [2], it is possible that 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. The source API exposing function takes the role of API invoker and communicates with the destination API exposing function over CAPIF-2.

4.3.4 API publishing function

The API publishing function (APF) enables the API provider to publish the Service APIs information using CAPIF-4 reference point as defined in 3GPP TS 23.222 [2] in order to enable the discovery of Service APIs by the API invoker.

4.3.5 API management function

The API management function (AMF) enables the API provider to perform administration of the Service APIs. The API management function supports several capabilities such as querying the Service API invocation log for auditing, monitoring the events, configuring the API provider policies and monitoring the status of the Service APIs using CAPIF-5 reference point as defined in in 3GPP TS 23.222 [2].

5 Services offered by the CAPIF Core Function

5.1 Introduction of Services

The table 5.1-1 lists the CAPIF Core Function APIs below the service name. A service description subclause for each API gives a general description of the related API.

Table 5.1-1: List of CAPIF Services

Service Name	Service Operations	Operation Semantics	Consumer(s)
CAPIF_Discover_Service_API	Discover_Service_API	Request/ Response	AP Invoker
	Event operations (NOTE)	(NOTE)	API Invoker
CAPIF_Publish_Service_API	Publish_Service_API	Request/ Response	API Publishing Function
	Unpublish_Service_API	Request/ Response	API Publishing Function
	Update_Service_API	Request/ Response	API Publishing Function
	Get_Service_API	Request/ Response	API Publishing Function
	Event operations (NOTE)	(NOTE)	API Publishing Function
CAPIF_Events_API	Subscribe_Event	Request/ Response	API Invoker, API Publishing Function, API Management Function, API Exposing Function
	Notify_Event	Subscribe/Notify	API Invoker, API Publishing Function, API Management Function, API Exposing Function
	Unsubscribe_Event	Request/ Response	API Invoker, API Publishing Function, API Management Function, API Exposing Function
CAPIF_API_Invoker_Management_API	Onboard_API_Invoker	Request/ Response	API Invoker
	Offboard_API_Invoker	Request/ Response	API Invoker
	Notify_Onboarding_Completion	Subscribe/Notify	API Invoker
CAPIF_Security_API	Obtain_Security_Method	Request/ Response	API Invoker
	Obtain_Authorization	Request/ Response	API Invoker
	Obtain_API_Invoker_Info	Request/ Response	API exposing function
	Revoke_Authorization	Request/ Response	API exposing function
CAPIF_Monitoring_API	Event operations (NOTE)	(NOTE)	API Management Function
CAPIF_Logging_API_Invocation_API	Log_API_Invocation	Request/ Response	API exposing function
CAPIF_Auditing_API	Query_API_Invocation_Log	Request/ Response	API management function
CAPIF_Access_Control_Policy_API	Obtain_Access_Control_Policy	Request/Response	API Exposing Function
NOTE: The service operations of CAPIF Events API are reused by the CAPIF_Discover_Service_API, CAPIF_Publish_Service_API and CAPIF_Monitoring_API for events related services.			

5.2 CAPIF_Discover_Service_API

5.2.1 Service Description

5.2.1.1 Overview

The CAPIF discover service APIs, as defined in 3GPP TS 23.222 [2], allow API invokers via CAPIF-1 and CAPIF-1e reference point to discover service API available at the CAPIF core function.

5.2.2 Service Operations

5.2.2.1 Introduction

The service operation defined for CAPIF_Discover_Service_API is shown in table 5.2.2.1-1.

Table 5.2.2.1-1: Operations of the CAPIF_Discover_Service_API

Service operation name	Description	Initiated by
Discover_Service_API	This service operation is used by an API invoker to discover service API available at the CAPIF core function.	API invoker

5.2.2.2 Discover_Service_API

5.2.2.2.1 General

This service operation is used by an API invoker to discover service API available at the CAPIF core function.

5.2.2.2.2 API invoker discovering service API using Discover_Service_API service operation

To discover service APIs available at the CAPIF core function, the API invoker shall send an HTTP GET message with the API invoker Identifier and query parameters to the CAPIF core function.

Upon receiving the above described HTTP GET message, the CAPIF core function shall:

1. verify the identity of the API invoker and check if the API invoker is authorized to discover the service APIs;
2. if the API invoker is authorized to discover the service APIs, the CAPIF core function shall:
   a. search the CAPIF core function (API registry) for APIs matching the query criteria;
   b. apply the discovery policy, if any, on the search results and filter the search results;
   c. return the filtered search results in the response message.

5.3 CAPIF_Publish_Service_API

5.3.1 Service Description

5.3.1.1 Overview

The CAPIF publish service APIs, as defined in 3GPP TS 23.222 [2], allow API publishing function via CAPIF-4 reference point to publish and manage published service APIs at the CAPIF core function.

5.3.2 Service Operations

5.3.2.1 Introduction

The service operations defined for the CAPIF_Publish_Service API are shown in table 5.3.2.1-1.

Table 5.3.2.1-1: Operations of the CAPIF_Publish_Service_API

Service operation name	Description	Initiated by
Publish_Service_API	This service operation is used by an API publishing function to publish service APIs on the CAPIF core function.	API publishing function
Unpublish_Service_API	This service operation is used by an API publishing function to un-publish service APIs from the CAPIF core function.	API publishing function
Get_Service_API	This service operation is used by an API publishing function to retrieve service APIs from the CAPIF core function.	API publishing function
Update_Service_API	This service operation is used by an API publishing function to update published service APIs on the CAPIF core function.	API publishing function

5.3.2.2 Publish_Service_API

5.3.2.2.1 General

This service operation is used by an API publishing function to publish service APIs on the CAPIF core function.

5.3.2.2.2 API publishing function publishing service APIs on CAPIF core function using Publish_Service_API service operation

To publish service APIs at the CAPIF core function, the API publishing function shall send an HTTP POST message to the CAPIF core function. The body of the HTTP POST message shall include API publishing function Identifier and API Information.

Upon receiving the above described HTTP POST message, the CAPIF core function shall:

1. verify the identity of the API publishing function and check if the API publishing function is authorized to publish service APIs;
2. if the API publishing function is authorized to publish service APIs, the CAPIF core function shall:
   a. verify the API Information present in the HTTP POST message and add the service APIs in the CAPIF core function (API registry);
   b. create a new resource as defined in subclause 8.2.3;
   c. send a notification message with the updated service API, to all API Invokers that subscribed to the Service API Update event; and
   d. return the CAPIF Resource URI in the response message.

5.3.2.3 Unpublish_Service_API

5.3.2.3.1 General

This service operation is used by an API publishing function to un-publish service APIs from the CAPIF core function.

5.3.2.3.2 API publishing function un-publishing service APIs from CAPIF core function using Unpublish_Service_API service operation

To un-publish service APIs from the CAPIF core function, the API publishing function shall send an HTTP DELETE message using the CAPIF Resource URI received during the publish operation to the CAPIF core function.

Upon receiving the above described HTTP DELETE message, the CAPIF core function shall

1. verify the identity of the API publishing function and check if the API publishing function is authorized to un-publish service APIs;
2. if the API publishing function is authorized to un-publish service APIs, the CAPIF core function shall:
   a. delete the resource pointed by the CAPIF Resource URI;
   b. delete the relevant service APIs from the CAPIF core function (API registry); and
   c. send a notification message with the deleted service API, to all API Invokers that subscribed to the Service API Update event.

5.3.2.4 Get_Service_API

5.3.2.4.1 General

This service operation is used by an API publishing function to retrieve service APIs from the CAPIF core function.

5.3.2.4.2 API publishing function retrieving service APIs from CAPIF core function using Get_Service_API service operation

To retrieve information about the published service APIs from the CAPIF core function, the API publishing function shall send an HTTP GET message with the API publishing function Identifier to the CAPIF core function.

Upon receiving the above described HTTP GET message, the CAPIF core function shall

1. verify the identity of the API publishing function and check if the API publishing function is authorized to retrieve information about the published service APIs;
2. if the API publishing function is authorized to retrieve information about the published service APIs, the CAPIF core function shall:
   a. respond with the API Information associated with the CAPIF Resource Identifier mentioned in the HTTP GET message.

5.3.2.5 Update_Service_API

5.3.2.5.1 General

This service operation is used by an API publishing function to update published service APIs on the CAPIF core function.

5.3.2.5.2 API publishing function updating published service APIs on CAPIF core function using Update_Service_API service operation

To update information of published service APIs, the API publishing function shall send an HTTP PUT message with the relevant CAPIF Resource URI and API publishing function Identifier to the CAPIF core function. The body of the HTTP PUT message shall include updated API Information.

Upon receiving the above described HTTP PUT message, the CAPIF core function shall

1. verify the identity of the API publishing function and check if the API publishing function is authorized to update information of published service APIs;
2. if the API publishing function is authorized to update information of published service APIs, the CAPIF core function shall:
   a. verify the API Information present in the HTTP PUT message and replace the service APIs in the CAPIF core function (API registry);
   b. replace the existing resource accordingly; and
   c. send a notification message with the updated service API, to all API Invokers that subscribed to the Service API Update event.

5.4 CAPIF_Events_API

5.4.1 Service Description

5.4.1.1 Overview

The CAPIF events APIs, as defined in 3GPP TS 23.222 [2], allow an API invoker via CAPIF-1 reference point, API exposure function via CAPIF-3 reference point, API publishing function via CAPIF-4 reference point and API management function via CAPIF-5 reference point to subscribe to and unsubscribe from CAPIF events and to receive notifications from CAPIF core function.

NOTE: The functional elements listed above are referred to as Subscriber in the service operations described in the subclauses below.

5.4.2 Service Operations

5.4.2.1 Introduction

The service operations defined for the CAPIF_Events_API are shown in table 5.4.2.1-1.

Table 5.4.2.1-1: Operations of the CAPIF_Events_API

Service operation name	Description	Initiated by
Subscribe_Event	This service operation is used by a Subscriber to subscribe to CAPIF events.	Subscriber
Unsubscribe_Event	This service operation is used by a Subscriber to unsubscribe from CAPIF events	Subscriber
Notify_Event	This service operation is used by CAPIF core function to send a notification to a Subscriber	CAPIF core function

5.4.2.2 Subscribe_Event

5.4.2.2.1 General

This service operation is used by a Subscriber to subscribe to CAPIF events.

5.4.2.2.2 Subscribing to CAPIF events using Subscribe_Event service operation

To subscribe to CAPIF events, the Subscriber shall send an HTTP POST message to the CAPIF core function. The body of the HTTP POST message shall include Subscriber's Identifier, Event Type and a Notification Destination URI.

Upon receiving the above described HTTP POST message, the CAPIF core function shall:

1. verify the identity of the Subscriber and check if the Subscriber is authorized to subscribe to the CAPIF events mentioned in the HTTP POST message;
2. if the Subscriber is authorized to subscribe to the CAPIF events, the CAPIF core function shall:
   a. create a new resource as defined in subclause 8.3.3; and
   b. return the CAPIF Resource URI in the response message.

5.4.2.3 Unsubscribe_Event

5.4.2.3.1 General

This service operation is used by a Subscriber to un-subscribe from CAPIF events.

5.4.2.3.2 Unsubscribing from CAPIF events using Unsubscribe_Event service operation

To unsubscribe from CAPIF events, the Subscriber shall send an HTTP DELETE message using the CAPIF Resource Identifier to the CAPIF core function.

Upon receiving the HTTP DELETE message, the CAPIF core function shall:

1. verify the identity of the Unsubscribing functional entity and check if the Unsubscribing functional entity is authorized to Unsubscribe from the CAPIF event associated with the CAPIF Resource URI; and
2. if the Unsubscribing functional entity is authorized to unsubscribe from the CAPIF events, the CAPIF core function shall delete the resource pointed by the CAPIF Resource URI.

5.4.2.4 Notify_Event

5.4.2.4.1 General

This service operation is used by CAPIF core function to send a notification to a Subscriber.

5.4.2.4.2 Notifying CAPIF events using Notify_Event service operation

To notify CAPIF events, the CAPIF core function shall send an HTTP POST message using the Notification Destination URI received in the subscription request. The body of the HTTP POST message shall include an Event Notification and CAPIF Resource URI.

Upon receiving the HTTP POST message, the Subscriber shall process the Event Notification.

5.5 CAPIF_API_Invoker_Management_API

5.5.1 Service Description

5.5.1.1 Overview

The CAPIF API invoker management APIs, as defined in 3GPP TS 23.222 [2], allow API invokers via CAPIF-1 and CAPIF-1e reference point to on-board and off-board itself as a recognized user of the CAPIF.

5.5.2 Service Operations

5.5.2.1 Introduction

The service operations defined for the CAPIF API Invoker Management API are shown in table 5.5.2.1-1.

Table 5.5.2.1-1: Operations of the CAPIF_API_Invoker_Management_API

Service operation name	Description	Initiated by
Onboard_API_Invoker	This service operation is used by an API invoker to on-board itself as a recognized user of CAPIF	API invoker
Offboard_API_Invoker	This service operation is used by an API invoker to off-board itself as a recognized user of CAPIF	API invoker
Notify_Onboarding_Completion	This service operation is used by CAPIF core function to send an on-boarding notification to the API invoker.	CAPIF core function

5.5.2.2 Onboard_API_Invoker

5.5.2.2.1 General

This service operation is used by an API invoker to on-board itself as a recognized user of CAPIF

5.5.2.2.2 API invoker on-boarding itself as a recognized user of CAPIF using Onboard_API_Invoker service operation

To on-board itself as a recognized user of the CAPIF, the API invoker shall send an HTTP POST message to the CAPIF core function. The body of the HTTP POST message shall include API invoker Enrolment Details, API List and a Notification Destination URI for on-boarding notification.

Upon receiving the above described HTTP POST message, the CAPIF core function shall check if it can determine authorization of the request and on-board the API invoker automatically. If the CAPIF core function:

1. can determine authorization of the request and on-board the API invoker automatically, the CAPIF core function:

   a. shall process the API invoker Enrolment Details and the API List received in the HTTP POST message and determine if the request sent by the API invoker is authorized or not;

   b. if the API invoker's request is authorized, the CAPIF core function shall:

      i. create the API invoker Profile consisting of an API invoker Identifier, Authentication Information, Authorization Information and CAPIF Identity Information;

      ii. verify the API List present in the HTTP POST message and create a API List of APIs the API invoker is allowed to access;

iii. create a new resource as defined in subclause 8.4.3;

iv. return the API invoker Profile, API List of APIs the API invoker is allowed to access and the CAPIF Resource URI in the response message.

2. cannot determine authorization of the request to on-board the API invoker automatically, the CAPIF core function:

   a. shall acknowledge the receipt of the on-boarding request to the API invoker.

   b. shall request the CAPIF administrator to validate the on-boarding request or the API management to validate the on-boarding request by sharing the API invoker Enrolment Details and the API List received in the HTTP POST message;

   c. on receiving confirmation of successful validation of the on-boarding request from the CAPIF administrator or the API management, the CAPIF core function shall:

      i. create the API invoker Profile consisting of an API invoker Identifier, Authentication Information, Authorization Information and CAPIF Identity Information;

      ii. create a new resource as defined in subclause 8.4.3;

      iii. deliver the API invoker Profile, API List of APIs the API invoker is allowed to access and the CAPIF Resource URI to the API invoker in a notification.

NOTE 1: How the CAPIF core function determines that the CAPIF core function can process the request and on-board the API invoker automatically is out-of-scope of this specification.

NOTE 2: How the CAPIF core function determines that the API invoker's request to on-board is authorized is specified in 3GPP TS 33.122 [16].

NOTE 3: Interactions between the CAPIF core function and the CAPIF administrator or the API management is out-of-scope of this specification.

NOTE 4: The onboarding credential received by the API invoker from the service provider as specified in 3GPP TS 33.122 [16] is included in the Authorization header field of the HTTP request message as described in IETF RFC 2617 [17].

NOTE 5: After the onboarding operation is completed the API Invoker no longer needs to maintain the Notification Destination URI and may delete it.

5.5.2.3 Offboard_API_Invoker

5.5.2.3.1 General

This service operation is used by an API invoker to stop being as a recognized user of CAPIF

5.5.2.3.2 API invoker off-boarding itself as a recognized user of CAPIF using Offboard_API_Invoker service operation

To off-board itself as a recognized user of the CAPIF, the API invoker shall send an HTTP DELETE message using the CAPIF Resource Identifier received during the on-boarding to the CAPIF core function.

Upon receiving the HTTP DELETE message, the CAPIF core function shall:

1. determine if the request sent by the API invoker is authorized or not;

2. if the API invoker's request is authorized, the CAPIF core function shall:

   a. delete the resource representation pointed by the CAPIF Resource Identifier; and

   b. delete the related API invoker profile.

5.5.2.4 Notify_Onboarding_Completion

5.5.2.4.1 General

This service operation is used by the CAPIF core function to send a notification about the completion of the Onboarding operation to the API Invoker.

5.5.2.4.2 Notifying Onboarding completion using Notify_Onboarding_Completion service operation

When the CAPIF core function cannot immediately authorize the API invoker that issued an Onboarding request (see subclause 5.5.2.2.2) it will send a response acknowledging the request and begin processing it. After completion, the CAPIF core function shall send an HTTP POST message using the Notification Destination URI received in the Onboarding request. The body of the HTTP POST message shall include the API Invoker Profile, API List of the APIs the API invoker is allowed to access and the CAPIF Resource URI.

Upon receiving the HTTP POST message, the API invoker shall process the message in the same manner it would have processed an immediate response to the Onboarding request, and respond to the HTTP POST message with an acknowledgement and no body.

5.6 CAPIF_Security_API

5.6.1 Service Description

5.6.1.1 Overview

The CAPIF security APIs, as defined in 3GPP TS 23.222 [2], allow:

- API invokers via CAPIF-1/1e reference points to negotiate the service security method and obtain authorization for invoking service APIs; and
- API exposing function via CAPIF-3 reference point to obtain authentication information of the API invoker for authentication of the API invoker.

5.6.2 Service Operations

5.6.2.1 Introduction

The service operations defined for CAPIF_Authentication_Authorization_API are shown in table 5.6.2.1-1.

Table 5.6.2.1-1: Operations of the CAPIF_Security_API

Service operation name	Description	Initiated by
Obtain_Security_Method	This service operation is used by an API invoker to negotiate and obtain information about service API security method for itself with CAPIF core function. This information is used by API invoker for service API invocations.	API invoker
Obtain_Authorization	This service operation is used by an API invoker to obtain authorization to access service APIs.	API invoker
Obtain_API_Invoker_Info	This service operation is used by an API exposing function to obtain the authentication or authorization information related to an API invoker.	API exposing function
Revoke_Authorization	This service operation is used by an API exposing function to invalidate the authorization of an API invoker.	API exposing function

5.6.2.2 Obtain_Security_Method

5.6.2.2.1 General

This service operation is used by an API invoker to negotiate and obtain service API security method from the CAPIF core function. The information received by API invoker shall be used for authentication with the API exposing function.

5.6.2.2.2 Request service API security method from CAPIF using Obtain_Security_Method service operation

To negotiate and obtain service API security method information from the CAPIF core function, the API invoker shall send an HTTP POST message to the CAPIF core function. The body of the HTTP POST message shall include Security Method Request and a Notification Destination URI for security related notifications. The Security Method Request from the API invoker contains the unique interface details of the service APIs and may contain a preferred method for each unique service API interface.

Upon receiving the above described HTTP POST message, the CAPIF core function shall:

1. determine the security method for each service API interface as specified in 3GPP TS 33.122 [16];
2. store the Notification Destination URI for security related notification;
3. create a new resource as defined in subclause 8.4.3; and
4. return the security method information and the CAPIF Resource URI in the response message.

5.6.2.3 Obtain_Authorization

5.6.2.3.1 General

This service operation is used by an API invoker to negotiate and obtain authorization information from the CAPIF core function. The information received by API invoker shall be used for authorization to invoke service APIs exposed by the API exposing function.

5.6.2.3.2 Obtain authorization using Obtain_Authorization service operation

To obtain authorization information from the CAPIF core function to invoke service APIs, the API invoker shall perform the functions of the resource owner, client and redirection endpoints as described in subclause 6.5.2.3 of 3GPP TS 33.122 [16].

5.6.2.4 Obtain_API_Invoker_Info

5.6.2.4.1 General

This service operation is used by an API exposing function to obtain the security information of API Invokers to be able to authenticate them and authorize each service API invocation by them.

5.6.2.4.2 Obtain API invoker's security information using Obtain_API_Invoker_Info service operation

To obtain authentication or authorization information from the CAPIF core function to authenticate or authorize an API invoker, the API exposing function shall send an HTTP GET message to the CAPIF core function with the API invoker ID and an indication to request authentication and authorization information.

Upon receiving the above described HTTP GET message, the CAPIF core function shall:

1. determine the security information of API invoker for all the service API interfaces of the API exposing function; and
2. return the security information in the response message.

5.6.2.5 Revoke_Authentication

5.6.2.5.1 General

This service operation is used by an API exposing function to invalidate the authorization of a specified API Invoker to invoke service APIs exposed by the calling API exposing function.

5.6.2.5.2 Invalidate authorization using Revoke_Authorization service operation

To invalidate authorization of an API invoker, the API exposing function shall send an HTTP DELETE message to the CAPIF core function using the API invoker ID.

Upon receiving the above described HTTP DELETE message, the CAPIF core function shall delete the resource representation pointed by the API invoker ID and shall notify the API invoker of the authorization invalidation using the Notification Destination URI received in the Obtain_Security_Method message.

5.7 CAPIF_Monitoring_API

The CAPIF monitoring API as defined in 3GPP TS 23.222 [2], allow the API management function via CAPIF-5 reference point to monitor service API invocations and receive such monitoring events from the CAPIF core function.

The CAPIF_Monitoring_API shall use the CAPIF_Events_API as described in subclause 8.3 by setting the CAPIFEvent to "Monitoring service API" as described in subclause 8.3.6.3.3.

5.8 CAPIF_Logging_API_Invocation_API

5.8.1 Service Description

5.8.1.1 Overview

The Logging API invocations APIs, as defined in 3GPP TS 23.222 [2], allow API exposing functions via CAPIF-3 reference point to log the information related to service API invocations on the CAPIF core function.

5.8.2 Service Operations

5.8.2.1 Introduction

Table 5.8.2.1-1: Operations of the CAPIF_Logging_API_Invocation_API

Service operation name	Description	Initiated by
Log_API_Invocation	This service operation is used by an API exposing function to log API invocation information on CAPIF core function.	API exposing function

5.8.2.2 Log_API_Invocation_API

5.8.2.2.1 General

This service operation is used by an API exposing function to log API invocation information on CAPIF core function.

5.8.2.2.2 Logging service API invocations using Log_API_Invocation service operation

To log service API invocations at the CAPIF core function, the API exposing function shall send an HTTP POST message to the CAPIF core function. The body of the HTTP POST message shall include API exposing function identity information and API invocation log information.

Upon receiving the above described HTTP POST message, the CAPIF core function shall:

1. verify the identity of the API exposing function and check if the API exposing function is authorized to create service API invocation logs;
2. if the API exposing function is authorized to create service API invocation logs, the CAPIF core function shall:
   a. process the API invocation log information received in the HTTP POST message and store the API invocation log information in the API repository;
   b. create a new resource as defined in subclause 8.7.3; and
   c. return the CAPIF Resource Identifier in the response message.

5.9 CAPIF_Auditing_API

5.9.1 Service Description

5.9.1.1 Overview

The Auditing API, as defined in 3GPP TS 23.222 [2], allows API management functions via CAPIF-5 reference point to query the log information stored on the CAPIF core function.

5.9.2 Service Operations

5.9.2.1 Introduction

Table 5.9.2.1-1: Operations of the CAPIF_Auditing_API

Service operation name	Description	Initiated by
Query_Invocation_Logs	This service operation is used by an API management function to query API invocation information logs stored on CAPIF core function.	API management function

5.9.2.2 Query_Invocation_Logs_API

5.9.2.2.1 General

This service operation is used by an API management function to query API invocation information logs stored on CAPIF core function.

5.9.2.2.2 Query API invocation information logs using Query_Invocation_Logs service operation

To query service API invocation logs at the CAPIF core function, the API management function shall send an HTTP GET message with the API management function identity information and the log query to the CAPIF core function.

Upon receiving the above described HTTP GET message, the CAPIF core function shall:

1. verify the identity of the API management function and check if the API management function is authorized to query the service API invocation logs;
2. if the API management function is authorized to query the service API invocation logs, the CAPIF core function shall:
   a. search the API invocation logs for logs matching the Log Query criteria; and
   b. return the search results in the response message.

5.10 CAPIF_Access_Control_Policy_API

5.10.1 Service Description

5.10.1.1 Overview

The CAPIF access control policy APIs allow API exposing function via CAPIF-3 reference point to obtain the service API access policy from the CAPIF core function.

5.10.2 Service Operations

5.10.2.1 Introduction

Table 5.3.2.1-1: Operations of the CAPIF_Access_Control_Policy_API

Service operation name	Description	Initiated by
Obtain_Access_Control_Policy	This service operation is used by an API exposing function to obtain the access control policy from the CAPIF core function.	API exposing function

5.10.2.2 Obtain_Access_Control_Policy

5.10.2.2.1 General

This service operation is used by an API exposing function to obtain the access control policy from the CAPIF core function.

5.10.2.2.2 API exposing function obtaining access control policy from the CAPIF core function using Obtain_Access_Control_Policy service operation

To obtain the access control policy from the CAPIF core function, the API exposing function shall send an HTTP GET message to the CAPIF core function with the API exposing function Identifier and API identification. The GET message may include API invoker ID for retrieving access control policy of the requested API invoker.

Upon receiving the above described HTTP GET message, the CAPIF core function shall

1. verify the identity of the API exposing function and check if the API exposing function is authorized to obtain the access control policy corresponding to the API identification;
2. if the API exposing function is authorized to obtain the access control policy, the CAPIF core function shall respond with the access control policy information corresponding to the API identification and API invoker ID (if present) in the HTTP GET message.

5.10.3 Related Events

The CAPIF_Access_Control_Policy_API supports the subscription and notification of the status of access control information via the CAPIF_Events_API. The related events are specified in subclause 8.3.6.3.3.

6 Services offered by the API exposing function

6.1 Introduction of Services

The table 6.1-1 lists the API exposing function APIs below the service name. A service description subclause for each API gives a general description of the related API.

Table 6.1-1: List of AEF Services

Service Name	Service Operations	Operation Semantics	Consumer(s)
AEF_Authentication_API	Authentication_Initiation_ Request	Request/ Response	API Invoker

6.2 AEF_Authentication_API

6.2.1 Service Description

6.2.1.1 Overview

The AEF authentication API, allows an API invokers via CAPIF-2/2e reference points to request API exposing function to ensure that authentication parameters necessary for authentication of the API invoker are available with the API exposing function. If the necessary authentication parameters are not available, the API exposing function fetches necessary authentication parameters from CAPIF core function to authenticate the API invoker.

6.2.2 Service Operations

6.2.2.1 Introduction

The service operation defined for AEF_Authentication _API is shown in table 6.2.2.1-1.

Table 6.2.2.1-1: Operations of the AEF_Authentication_API

Service operation name	Description	Initiated by
Authentication_Initiation_Request	This service operation is used by an API invoker to request API exposing function to fetch necessary authentication parameters from CAPIF core function to authenticate the API invoker	API invoker

6.2.2.2 Authentication_Initiation_Request_API

6.2.2.2.1 General

This service operation is used by an API invoker to initiate authentication with the API exposing function. On receiving the Authentication_Initiation_Request the API exposing function fetches the authentication information of the API invoker from the CAPIF core function, if required.

6.2.2.2.2 API invoker initiating authentication using Authentication_Initiation_Request service operation

To initiate authentication with the API exposing function, the API invoker shall send an HTTP GET message to the API exposing function with the API invoker ID.

Upon receiving the above described HTTP GET message, the API exposing function shall check if the credentials of the API invoker for authentication are available with the API exposing function. If the credentials of the API invoker for authentication are not available, the API exposing function shall use the service defined in subclause 5.6.2.4.2 to fetch the credentials from the CAPIF core function.

7 CAPIF Design Aspects Common for All APIs

7.1 General

CAPIF APIs are RESTful APIs that allow secure access to the capabilities provided by CAPIF.

This document specifies the procedures triggered at different functional entities as a result of API invocation requests and event notifications. The stage-2 level requirements and signalling flows are defined in 3GPP TS 23.222 [2].

Several design aspects, as mentioned in the following subclauses, are specified in 3GPP TS 29.122 [14] and referenced by this specification.

7.2 Data Types

7.2.1 General

This clause defines structured data types, simple data types and enumerations that are applicable to several APIs defined in the present specification and can be referenced from data structures defined in the subsequent clauses.

In addition, data types that are defined in OpenAPI 3.0.0 Specification [3] can also be referenced from data structures defined in the subsequent clauses.

NOTE: As a convention, data types in the present specification are written with an upper-case letter in the beginning. Parameters are written with a lower-case letter in the beginning. As an exception, data types that are also defined in OpenAPI 3.0.0 Specification [3] can use a lower-case case letter in the beginning for consistency.

Table 7.2.1-1 specifies data types re-used by the CAPIF from other specifications, including a reference to their respective specifications and when needed, a short description of their use within the CAPIF.

Table 7.2.1-1: Re-used Data Types

Data type	Reference	Comments
Uri	3GPP TS 29.122 [14]	
TestNotification	3GPP TS 29.122 [14]	Following clarifications apply: - The SCEF is the CAPIF core function; and - The SCS/AS is theSubscriber.
WebsockNotifConfig	3GPP TS 29.122 [14]	Following clarifications apply: - The SCEF is the CAPIF core function; and - The SCS/AS is theSubscriber.

7.2.2 Referenced structured data types

Table 7.2.2-1 lists structured data types defined in this specification referenced by multiple services:

Table 7.2.2-1: Referenced Structured Data Types

Data type	Reference	Description
Log	Subclause 8.7.4.2.3	Individual log entries
InterfaceDescription	Subclause 8.2.4.2.3	Description of the API interface
ServiceAPIDescription	Subclause 8.2.4.2.2	Description of the service API

7.2.3 Referenced Simple data types and enumerations

Following simple data types defined in Table 7.2.3.1-1 are applicable to several APIs in this document:

Table 7.2.3.1-1: Simple data types applicable to several APIs

Type name	Reference	Description
CAPIFResourceId	n/a	string chosen by the CAPIF core function to serve as identifier in a resource URI.
DataFormat	Subclause 8.2.4.3.4	Data format used by the API
Protocol	Subclause 8.2.4.3.3	Protocol used by the API

7.3 Usage of HTTP

For CAPIF APIs, support of HTTP/1.1 (IETF RFC 7230 [4], IETF RFC 7231 [5], IETF RFC 7232 [6], IETF RFC 7233 [7], IETF RFC 7234 [8] and IETF RFC 7235 [9]) over TLS (IETF RFC 5246 [11]) is mandatory and support of HTTP/2 (IETF RFC 7540 [10]) over TLS (IETF RFC 5246 [11]) is recommended.

A functional entity desiring to use HTTP/2 shall use the HTTP upgrade mechanism to negotiate applicable HTTP version as described in IETF RFC 7540 [10].

7.4 Content type

The bodies of HTTP request and successful HTTP responses shall be encoded in JSON format (see IETF RFC 7159 [12]).

The MIME media type that shall be used within the related Content-Type header field is "application/json", as defined in IETF RFC 7159 [12].

NOTE: This release only supports the content type JSON.

7.5 URI structure

All resource URIs of CAPIF APIs should have the following root structure:

{apiRoot}/{apiName}/{apiVersion}/

"apiRoot" is configured by means outside the scope of the present document. It includes the scheme ("https"), host and optional port, and an optional prefix string. "apiName" and "apiVersion" shall be set dependent on the API, as defined in the corresponding subclauses below.

All resource URIs in the subclauses below are defined relative to the above root URI.

NOTE 1: The "apiVersion" will only be increased if the new API version contains backward incompatible changes. Otherwise, the supported feature mechanism defined in subclause 7.8 can be used to negotiate extensions.

NOTE 2: A different root structure can be used when the resource URI is preconfigured in the API invoking entity.

The root structure may be followed by "apiSpecificSuffixes" that are dependent on the API and are defined separately for each API where they apply:

{apiRoot}/{apiName}/{apiVersion}/{apiSpecificSuffixes}

7.6 Notifications

The functional entities

- shall support the delivery of notifications using a separate HTTP connection towards an address;
- may support testing delivery of notifications; and
- may support the delivery of notification using WebSocket protocol (see IETF RFC 6455 [13]),

as described in 3GPP TS 29.122 [14], with the following clarifications:

- the SCEF is the CAPIF core function; and
- the SCS/AS is theSubscriber.

7.7 Error handling

Response bodies for error handling, as described in 3GPP TS 29.122 [14], are applicable to all APIs in the present specification unless specified otherwise, with the following clarifications:

- the SCEF is the CAPIF core function; and
- the SCS/AS is the functional entity invoking an API.

7.8 Feature negotiation

The functional entity invoking an API (i.e. the API invoker, the API exposing function, the API publishing function or the API management function) and the CAPIF core function use feature negotiation procedures defined in 3GPP TS 29.122 [14] to negotiate the supported features, with the following clarifications:

- The SCEF is the CAPIF core function; and

- The SCS/AS is the functional entity invoking an API.

7.9 HTTP headers

The HTTP headers described in 3GPP TS 29.122 [14] are applicable to all APIs in this document.

8 CAPIF API Definition

8.1 CAPIF_Discover_Service_API

8.1.1 API URI

The request URI used in each HTTP request from the API invoker towards the CAPIF core function shall have the structure as defined in subclause 7.5 with the following clarifications:

- The {apiName} shall be "service-apis".
- The {apiVersion} shall be "v1".
- The {apiSpecificSuffixes} shall be set as described in subclause 8.1.2.

8.1.2 Resources

8.1.2.1 Overview

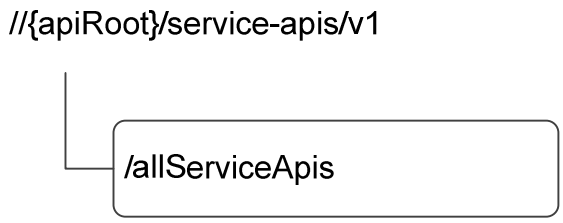

Figure 8.1.2.1-1: Resource URI structure of the CAPIF_Discover_Service_API

Table 8.1.2.1-1 provides an overview of the resources and applicable HTTP methods.

Table 8.1.2.1-1: Resources and methods overview

Resource name	Resource URI	HTTP method or custom operation	Description
All published service APIs (Store)	{apiRoot} /service-apis/v1 /allServiceApis	GET	Discover published service APIs and retrieve a collection of APIs according to certain filter criteria.

8.1.2.2 Resource: All published service APIs

8.1.2.2.1 Description

The All published service APIs resource represents a collection of published service APIs on a CAPIF core function. The resource is modelled as a Store resource archetype (see Annex C.3 of 3GPP TS 29.501 [18])

8.1.2.2.2 Resource Definition

Resource URI: **{apiRoot}/service-apis/v1/allServiceApis**

This resource shall support the resource URI variables defined in table 8.1.2.2.2-1.

Table 8.1.2.2.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See subclause 7.5

8.1.2.2.3 Resource Standard Methods

8.1.2.2.3.1 GET

This operation retrieves a list of APIs currently registered in the CAPIF core function, satisfying a number of filter criteria.

Table 8.1.2.2.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	P	Cardinality	Description
apiInvokerId	string	M	1	String identifying the API invoker assigned by the CAPIF core function.
serviceName	string	O	0..1	Name of the service
apiName	string	O	0..1	API name
apiVersion	string	O	0..1	API version
commType	CommunicationType	O	0..1	Communication type used by the API (e.g. request/response or subscribe/notify).
interfaceDescription	array(InterfaceDescription)	O	0..N	Interface details (e.g. domain name, ipv4/6Addr, port).
dataFormat	DataFormat	O	0..1	Data format used by the API (e.g. serialization protocol JSON used).

This method shall support the request data structures specified in table 8.1.2.2.3.1-2 and the response data structures and response codes specified in table 8.1.2.2.3.1-3.

Table 8.1.2.2.3.1-2: Data structures supported by the GET Request Body on this resource

Data type	P	Cardinality	Description
n/a			

Table 8.1.2.2.3.1-3: Data structures supported by the GET Response Body on this resource

Data type	P	Cardinality	Response codes	Description
array(DiscoveredAPIs)	O	1..N	200 OK	The response body contains the result of the search over the list of registered APIs.
ProblemDetails	M	1	414 URI Too Long	Indicates that the server is refusing to service the request because the request-target is too long.

8.1.2.2.4 Resource Custom Operations

None.

8.1.3 Notifications

None.

8.1.4 Data Model

8.1.4.1 General

This subclause specifies the application data model supported by the API.

Table 8.1.4.1-1 specifies the data types defined for the CAPIF service based interface protocol.

Table 8.1.4.1-1: Specific Data Types

Data type	Section defined	Description	Applicability
DiscoveredAPIs	8.1.4.2.2	Definition of the service API	

Table 8.1.4.1-2 specifies data types re-used by the CAPIF_Discover_Service_API service:

Table 8.1.4.1-2: Re-used Data Types

Data type	Reference	Comments	Applicability
ServiceAPIDescription	Subclause 8.2.4.2.2	Description of the service API	
InterfaceDescription	Subclause 8.2.4.2.3	Name of the protocol	
Protocol	Subclause 8.2.4.3.3	Protocol	
DataFormat	Subclause 8.2.4.3.4	Data format	
CommunicationType	Subclause 8.2.4.3.5	Communication type used by the API	
Uri	3GPP TS 29.122 [14]		
ProblemDetails	3GPP TS 29.122 [14]		

8.1.4.2 Structured data types

8.1.4.2.1 Introduction

8.1.4.2.2 Type: DiscoveredAPIs

Table 8.1.4.2.2-1: Definition of type DiscoveredAPIs

Attribute name	Data type	P	Cardinality	Description	Applicability
apiId	string	M	1	Identifier of the service API	
serviceAPIDescription	ServiceAPIDescription	M	1	Description of the service API as published by the service.	

8.1.4.3 Simple data types and enumerations

None.

8.1.5 Error Handling

General error responses are defined in subclause 7.7.

8.1.6 Feature negotiation

General feature negotiation procedures are defined in subclause 7.8.

Table 8.1.6-1: Supported Features

Feature number	Feature Name	Description
n/a		

8.2 CAPIF_Publish_Service_API

8.2.1 API URI

The request URI used in each HTTP request from the API publishing function towards the CAPIF core function shall have the structure as defined in subclause 7.5 with the following clarifications:

- The {apiName} shall be "published-apis".
- The {apiVersion} shall be "v1".

- The {apiSpecificSuffixes} shall be set as described in subclause 8.2.2.

8.2.2 Resources

8.2.2.1 Overview

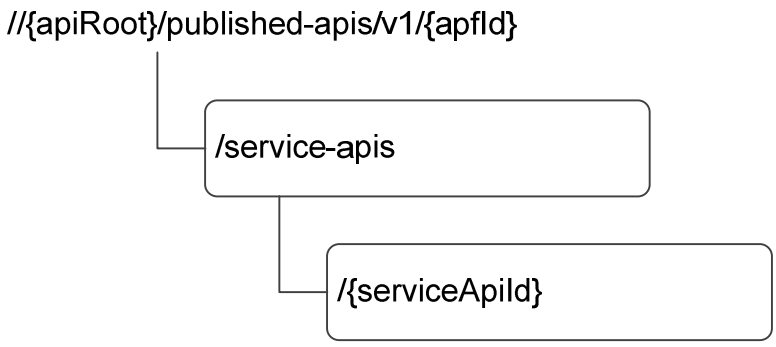

Figure 8.2.2.1-1: Resource URI structure of the CAPIF_Publish_Service_API

Table 8.2.2.1-1 provides an overview of the resources and applicable HTTP methods.

Table 8.2.2.1-1: Resources and methods overview

Resource name	Resource URI	HTTP method or custom operation	Description
APF published APIs	{apiRoot} /published-apis/v1 /{apfId}/service-apis	POST	Publish a new API
		GET	Retrieve all published service APIs
Individual APF published API	{apiRoot} /published-apis /v1 /{apfId}/service-apis /{serviceApiId}	GET	Retrieve a published service API
		PUT	Update a published service API
		DELETE	Unpublish a published service API

8.2.2.2 Resource: APF published APIs

8.2.2.2.1 Description

The APF published APIs resource represents all published service APIs of a API publishing function.

8.2.2.2.2 Resource Definition

Resource URI: **{apiRoot}/published-apis/v1/{apfId}/service-apis**

This resource shall support the resource URI variables defined in table 8.2.2.2.2-1.

Table 8.2.2.2.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See subclause 7.5
apfId	String identifying the API publishing function

8.2.2.2.3 Resource Standard Methods

8.2.2.2.3.1 POST

This method shall support the URI query parameters specified in table 8.2.2.2.3.1-1.

Table 8.2.2.2.3.1-1: URI query parameters supported by the POST method on this resource

Name	Data type	P	Cardinality	Description
n/a				

This method shall support the request data structures specified in table 8.2.2.2.3.1-2 and the response data structures and response codes specified in table 8.2.2.2.3.1-3.

Table 8.2.2.2.3.1-2: Data structures supported by the POST Request Body on this resource

Data type	P	Cardinality	Description
ServiceAPIDescription	M	1	Definition of the service API being published

Table 8.2.2.2.3.1-3: Data structures supported by the POST Response Body on this resource

Data type	P	Cardinality	Response codes	Description
ServiceAPIDescription	M	1	201 Created	Service API published successfully. The URI of the created resource shall be returned in the "Location" HTTP header

8.2.2.2.3.2 GET

This method shall support the URI query parameters specified in table 8.2.2.2.3.2-1.

Table 8.2.2.2.3.2-1: URI query parameters supported by the GET method on this resource

Name	Data type	P	Cardinality	Description
n/a				

This method shall support the request data structures specified in table 8.2.2.2.3.2-2 and the response data structures and response codes specified in table 8.2.2.2.3.2-3.

Table 8.2.2.2.3.2-2: Data structures supported by the GET Request Body on this resource

Data type	P	Cardinality	Description
n/a			

Table 8.2.2.2.3.1-3: Data structures supported by the GET Response Body on this resource

Data type	P	Cardinality	Response codes	Description
array(ServiceAPIDescription)	O	0..N	200 OK	Definition of all service API(s) published by the API publishing function.

8.2.2.2.4 Resource Custom Operations

None.

8.2.2.3 Resource: Individual APF published API

8.2.2.3.1 Description

The Individual APF published API resource represents an individual published service API.

8.2.2.3.2 Resource Definition

Resource URI: **{apiRoot}/published-apis/v1/{apfId}/service-apis/{serviceApiId}**

This resource shall support the resource URI variables defined in table 8.2.2.3.2-1.

Table 8.2.2.3.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See subclause 7.5
apfId	String identifying the API publishing function
serviceApiId	String identifying an individual published service API

8.2.2.3.3 Resource Standard Methods

8.2.2.3.3.1 GET

This method shall support the URI query parameters specified in table 8.2.2.3.3.1-1.

Table 8.2.2.3.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	P	Cardinality	Description
n/a				

This method shall support the request data structures specified in table 8.2.2.3.3.1-2 and the response data structures and response codes specified in table 8.2.2.3.3.1-3.

Table 8.2.2.3.3.1-2: Data structures supported by the GET Request Body on this resource

Data type	P	Cardinality	Description
n/a			

Table 8.2.2.3.3.1-3: Data structures supported by the GET Response Body on this resource

Data type	P	Cardinality	Response codes	Description
array(ServiceAPIDescription)	O	0..N	200 OK	Definition of all service API published by the API publishing function.

8.2.2.3.3.2 PUT

This method shall support the URI query parameters specified in table 8.2.2.3.3.2-1.

Table 8.2.2.3.3.2-1: URI query parameters supported by the PUT method on this resource

Name	Data type	P	Cardinality	Description
n/a				

This method shall support the request data structures specified in table 8.2.2.3.3.2-2 and the response data structures and response codes specified in table 8.2.2.3.3.2-3.

Table 8.2.2.3.3.2-2: Data structures supported by the PUT Request Body on this resource

Data type	P	Cardinality	Description
ServiceAPIDescription	M	1	Updated definition of the service API.

Table 8.2.2.3.3.2-3: Data structures supported by the PUT Response Body on this resource

Data type	P	Cardinality	Response codes	Description
ServiceAPIDescription	M	1	200 OK	Definition of the service API updated successfully..

8.2.2.3.3.3 DELETE

This method shall support the URI query parameters specified in table 8.2.2.3.3.3-1.

Table 8.2.2.3.3.3-1: URI query parameters supported by the GET method on this resource

Name	Data type	P	Cardinality	Description
n/a				

This method shall support the request data structures specified in table 8.2.2.3.3.3-2 and the response data structures and response codes specified in table 8.2.2.3.3.3-3.

Table 8.2.2.3.3.3-2: Data structures supported by the DELETE Request Body on this resource

Data type	P	Cardinality	Description
n/a			

Table 8.2.2.3.3.3-3: Data structures supported by the DELETE Response Body on this resource

Data type	P	Cardinality	Response codes	Description
n/a			204 No Content	The individual published service API matching the serviceApiId is deleted.

8.2.2.3.4 Resource Custom Operations

None.

8.2.3 Notifications

None.

8.2.4 Data Model

8.2.4.1 General

This subclause specifies the application data model supported by the API.

Table 8.2.4.1-1 specifies the data types defined for the CAPIF service based interface protocol.

Table 8.2.4.1-1: Specific Data Types

Data type	Section defined	Description	Applicability
ServiceAPIDescription	8.2.4.2.2	Definition of the service API	
InterfaceDescription	8.2.4.2.3	Description of the API interface	

Table 8.2.4.1-2 specifies data types re-used by the CAPIF_Publish_Service_API service:

Table 8.2.4.1-2: Re-used Data Types

Data type	Reference	Comments	Applicability
Ipv4Addr	3GPP TS 29.122 [14]		
Ipv6Addr	3GPP TS 29.122 [14]		
Uri	3GPP TS 29.122 [14]		
Port	3GPP TS 29.122 [14]		

8.2.4.2 Structured data types

8.2.4.2.1 Introduction

8.2.4.2.2 Type: ServiceAPIDescription

Table 8.2.4.2.2-1: Definition of type ServiceAPIDescription

Attribute name	Data type	P	Cardinality	Description	Applicability
apiName	string	M	1	API name	
apiID	string	C	0..1	API identifier assigned by the CAPIF core function to the published service API. Shall not be present in the HTTP POST request from the API publishing function to the CAPIF core function. Shall be present in the HTTP POST response from the CAPIF core function to the API publishing function.	
apiVersion	string	O	0..1	API version	
domainName	string	O	0..1	Domain to which API belongs to	
protocol	Protocol	O	0..1	Protocol used by the API.	
serviceName	string	O	0..1	Name of the service to which the API belongs	
commType	CommunicationType	O	0..1	Communication type used by the API	
interfaceDescriptions	array(InterfaceDescription)	O	0..N	Interface details	
dataFormat	DataFormat	O	0..1	Data formats used by the API	
description	string	O	0..1	Text description of the API	
uris	array(Uri)	O	0..N	Relative URI (s) of the API	

8.2.4.2.3 Type: InterfaceDescription

Table 8.2.4.2.3-1: Definition of type InterfaceDescription

Attribute name	Data type	P	Cardinality	Description	Applicability
ipv4Addr	Ipv4Addr	O	0..1	String identifying an IPv4 address	
ipv6Addr	Ipv6Addr	O	0..1	String identifying an IPv6 address	
port	Port	O	0..1	Port	
securityMethods	array(SecurityMethods)	M	1..N	Security methods supported by the interface	

8.2.4.3 Simple data types and enumerations

8.2.4.3.1 Introduction

This subclause defines simple data types and enumerations that can be referenced from data structures defined in the previous subclauses.

8.2.4.3.2 Simple data types

The simple data types defined in table 8.2.4.3.2-1 shall be supported.

Table 8.2.4.3.2-1: Simple data types

Type Name	Type Definition	Description	Applicability
n/a			

8.2.4.3.3 Enumeration: Protocol

Table 8.2.4.3.3-1: Enumeration Protocol

Enumeration value	Description	Applicability
HTTP_1_1	HTTP version 1.1	
HTTP2	HTTP version 2	

8.2.4.3.4 Enumeration: DataFormat

Table 8.2.4.3.4-1: Enumeration DataFormat

Enumeration value	Description	Applicability
JSON	Serialization protocol: JavaScript Object Notation	

8.2.4.3.5 Enumeration: CommunicationType

Table 8.2.4.3.5-1: Enumeration CommunicationType

Enumeration value	Description	Applicability
REQUEST_RESPONSE	The communication is of the type request-response.	
SUBSCRIBE_NOTIFY	The communication is of the type subscribe-notify	

8.2.4.3.6 Enumeration: SecurityMethods

Table 8.2.4.3.6-1: Enumeration SecurityMethods

Enumeration value	Description	Applicability
PSK	Security method 1 (Using TLS-PSK) as described in 3GPP TS 33.122 [16].	
PKI	Security method 2 (Using PKI) as described in 3GPP TS 33.122 [16].	
OAUTH	Security method 3 (TLS with OAuth token) as described in 3GPP TS 33.122 [16].	

8.2.5 Error Handling

General error responses are defined in subclause 7.7.

8.2.6 Feature negotiation

General feature negotiation procedures are defined in subclause 7.8.

Table 8.2.6-1: Supported Features

Feature number	Feature Name	Description
n/a		

8.3 CAPIF_Events_API

8.3.1 API URI

The request URI used in each HTTP request from the Subscriber towards the CAPIF core function shall have the structure as defined in subclause 7.5 with the following clarifications:

- The {apiName} shall be "capif-events".
- The {apiVersion} shall be "v1".
- The {apiSpecificSuffixes} shall be set as described in subclause 8.3.2.

8.3.2 Resources

8.3.2.1 Overview

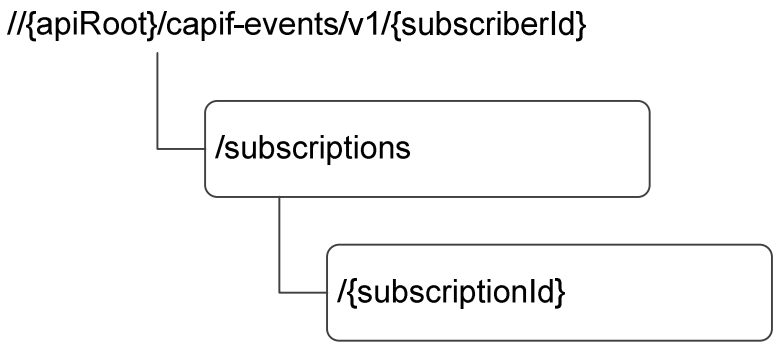

Figure 8.3.2.1-1: Resource URI structure of the CAPIF_Events_API

Table 8.3.2.1-1 provides an overview of the resources and applicable HTTP methods.

Table 8.3.2.1-1: Resources and methods overview

Resource name	Resource URI	HTTP method or custom operation	Description
CAPIF Events Subscriptions	{apiRoot}/capif-events/v1/{subscriberId}/subscriptions	POST	Creates a new individual CAPIF Event Subscription
Individual CAPIF Events Subscription	{apiRoot}/capif-events/v1/{subscriberId}/subscriptions/{subscriptionId}	DELETE	Deletes an individual CAPIF Event Subscription identified by the subscriptionId

8.3.2.2 Resource: CAPIF Events Subscriptions

8.3.2.2.1 Description

The CAPIF Events Subscriptions resource represents all subscriptions of aSubscriber.

8.3.2.2.2 Resource Definition

Resource URI: {apiRoot}/capif-events/v1/{subscriberId}/subscriptions

This resource shall support the resource URI variables defined in table 8.3.2.2.2-1.

Table 8.3.2.2.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See subclause 7.5
subscriberId	ID of the Subscriber

8.3.2.2.3 Resource Standard Methods

8.3.2.2.3.1 POST

This method shall support the URI query parameters specified in table 8.3.2.2.3.1-1.

Table 8.3.2.2.3.1-1: URI query parameters supported by the POST method on this resource

Name	Data type	P	Cardinality	Description
n/a				

This method shall support the request data structures specified in table 8.3.2.2.3.1-2 and the response data structures and response codes specified in table 8.3.2.2.3.1-3.

Table 8.3.2.2.3.1-2: Data structures supported by the POST Request Body on this resource

Data type	P	Cardinality	Description
EventSubscription	M	1	Create a new individual CAPIF Events Subscription resource.

Table 8.3.2.2.3.1-3: Data structures supported by the POST Response Body on this resource

Data type	P	Cardinality	Response codes	Description
EventSubscription	M	1	201 Created	CAPIF Events Subscription resource created successfully. The URI of the created resource shall be returned in the "Location" HTTP header

8.3.2.2.4 Resource Custom Operations

None.

8.3.2.3 Resource: Individual CAPIF Events Subscription

8.3.2.3.1 Description

The Individual CAPIF Events Subscription resource represents an individual event subscription of aSubscriber.

8.3.2.3.2 Resource Definition

Resource URI: {apiRoot}/capif-events/v1/{subscriberId}/subscriptions/{subscriptionId}

This resource shall support the resource URI variables defined in table 8.3.2.3.2-1.

Table 8.3.2.3.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See subclause 7.5
subscriberId	ID of the Subscriber
subscriptionId	String identifying an individual Events Subscription

8.3.2.3.3 Resource Standard Methods

8.3.2.3.3.1 DELETE

This method shall support the URI query parameters specified in table 8.3.2.3.3.1-1.

Table 8.3.2.3.3.1-1: URI query parameters supported by the DELETE method on this resource

Name	Data type	P	Cardinality	Description
n/a				

This method shall support the request data structures specified in table 8.3.2.3.3.1-2 and the response data structures and response codes specified in table 8.3.2.3.3.1-3.

Table 8.3.2.3.3.1-2: Data structures supported by the DELETE Request Body on this resource

Data type	P	Cardinality	Description
n/a			

Table 8.3.2.3.3.1-3: Data structures supported by the DELETE Response Body on this resource

Data type	P	Cardinality	Response codes	Description
n/a			204 No Content	The individual CAPIF Events Subscription matching the subscriptionId is deleted.

8.3.2.3.4 Resource Custom Operations

None.

8.3.3 Notifications

8.3.3.1 General

The delivery of notifications shall conform to subclause 7.6.

Table 8.3.3.1-1: Notifications overview

Notification	Resource URI	HTTP method or custom operation	Description (service operation)
Event notification	{notificationDestination}	POST	Notifies Subscriber of a CAPIF Event

8.3.3.2 Event Notification

8.3.3.2.1 Description

Event Notification is used by the CAPIF core function to notify a Subscriber of an Event. The Subscriber shall be subscribed to such Event Notification via the Individual CAPIF Events Subscription Resource.

8.3.3.2.2 Notification definition

The POST method shall be used for Event notification and the URI shall be the one provided by the Subscriber during the on-boarding request.

Resource URI: **{notificationDestination}**

This method shall support the URI query parameters specified in table 8.3.3.2.2.1-1.

Table 8.3.3.2.2-1: URI query parameters supported by the POST method on this resource

Name	Data type	P	Cardinality	Description
n/a				

This method shall support the request data structures specified in table 8.3.3.2.2-2 and the response data structures and response codes specified in table 8.3.3.2.2-3.

Table 8.3.3.2.2-2: Data structures supported by the POST Request Body on this resource

Data type	P	Cardinality	Description
EventNotification	M	1	Notification information of a CAPIF Event

Table 8.3.3.2.2-3: Data structures supported by the POST Response Body on this resource

Data type	P	Cardinality	Response codes	Description
n/a			204 No Content	The receipt of the Notification is acknowledged.

8.3.4 Data Model

8.3.4.1 General

This subclause specifies the application data model supported by the API. Data types listed in subclause 7.2 apply to this API.

Table 8.3.4.1-1 specifies the data types defined specifically for the CAPIF_Events_API service.

Table 8.3.4.1-1: CAPIF_Events_API specific Data Types

Data type	Section defined	Description	Applicability
EventSubscription	8.3.4.2.2	Represents an individual CAPIF Event Subscription resource	
EventNotification	8.3.4.2.3	Represents an individual CAPIF Event Subscription Notification resource	
CAPIFEvent	8.3.4.3.2	Describes CAPIF events	

Table 8.3.4.1-2 specifies data types re-used by the CAPIF_Events_API service:

Table 8.3.4.1-2: Re-used Data Types

Data type	Reference	Comments	Applicability
Uri	3GPP TS 29.122 [14]		
TestNotification	3GPP TS 29.122 [14]	Following differences apply: - The SCEF is the CAPIF core function; and - The SCS/AS is the Subscribing functional entity.	
WebsockNotifConfig	3GPP TS 29.122 [14]	Following differences apply: - The SCEF is the CAPIF core function; and - The SCS/AS is the Subscribing functional entity.	

8.3.4.2 Structured data types

8.3.4.2.1 Introduction

This subclause defines the structures to be used in resource representations.

8.3.4.2.2 Type: EventSubscription

Table 8.3.4.2.2-1: Definition of type EventSubscription

Attribute name	Data type	P	Cardinality	Description	Applicability
events	array(CAPIFEvent)	M	1..N	Subscribed events	
notificationDestination	Uri	M	1	URI where the notification should be delivered to.	
requestTestNotification	boolean	O	0..1	Set to true by Subscriber to request the CAPIF core function to send a test notification as defined in subclause 7.6. Set to false or omitted otherwise.	Notification_test_event
websockNotifConfig	WebsockNotifConfig	O	0..1	Configuration parameters to set up notification delivery over Websocket protocol as defined in subclause 7.6.	Notification_websocket

8.3.4.2.3 Type: EventNotification

Table 8.3.4.2.4-1: Definition of type EventNotification

Attribute name	Data type	P	Cardinality	Description	Applicability
subscriptionId	string	M	1	Identifier of the subscription resource to which the notification is related – CAPIF resource identifier	
events	array(CAPIFEvent)	M	1..N	Notifications of individual events	

8.3.4.3 Simple data types and enumerations

8.3.4.3.1 Introduction

This subclause defines simple data types and enumerations that can be referenced from data structures defined in the previous subclauses.

8.3.4.3.2 Simple data types

None.

The simple data types defined in table 8.3.4.3.2-1 shall be supported.

Table 8.3.4.3.2-1: Simple data types

Type Name	Type Definition	Description	Applicability
n/a			

8.3.4.3.3 Enumeration: CAPIFEvent

Table 8.3.4.3.3-1: Enumeration CAPIFEvent

Enumeration value	Description	Applicability
SERVICE_API_AVAILABLE	Events related to the availability of service APIs after the service APIs are published.	
SERVICE_API_UNAVAILABLE	Events related to the unavailability of service APIs after the service APIs are unpublished.	
SERVICE_API_UPDATE	Events related to change in service API information	
API_INVOKER_ONBOARDED	Events related to API invoker onboarded to CAPIF	
API_INVOKER_OFFBOARDED	Events related to API invoker offboarded from CAPIF	
SERVICE_API_INVOCATION_SUCCESS	Events related to the successful invocation of service APIs	
SERVICE_API_INVOCATION_FAILURE	Events related to the failed invocation of service APIs	
ACCESS_CONTROL_POLICY_UPDATE	Events related to the update for the access control policy related to the service APIs	
ACCESS_CONTROL_POLICY_UNAVAILABLE	Events related to the unavailability of the access control policy related to the service APIs	
API_INVOKER_AUTHORIZATION_REVOKED	Events related to the revocation of the authorization of API invokers to access the service APIs.	

8.3.5 Error Handling

General error responses are defined in subclause 7.7.

8.3.6 Feature negotiation

General feature negotiation procedures are defined in subclause 7.8. Table 8.3.6-1 lists the supported features for CAPIF_Events_API.

Table 8.3.6-1: Supported Features

Feature number	Feature Name	Description
1	Notification_test_event	Testing of notification connection is supported according to subclause 7.6.
2	Notification_websocket	The delivery of notifications over Websocket is supported according to subclause 7.6. This feature requires that the Notification_test_event feature is also supported.

Editor's Note: Supporting features specific to Subscriber is for further study.

8.4 CAPIF_API_Invoker_Management_API

8.4.1 API URI

The request URI used in each HTTP request from the API invoker towards the CAPIF core function shall have the structure as defined in subclause 7.5 with the following clarifications:

- The {apiName} shall be "api-invoker-management".
- The {apiVersion} shall be "v1".
- The {apiSpecificSuffixes} shall be set as described in subclause 8.4.2.

8.4.2 Resources

8.4.2.1 Overview

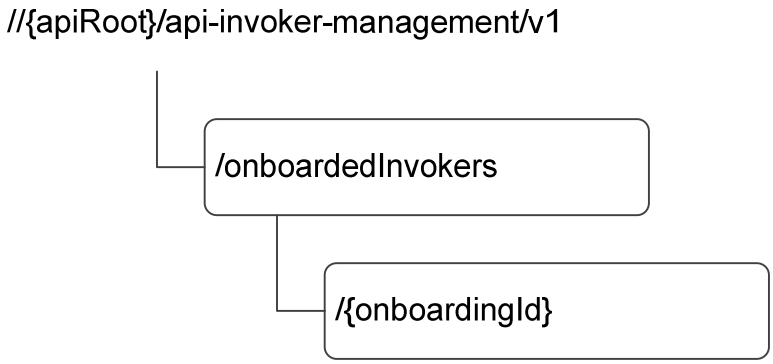

Figure 8.4.2.1-1: Resource URI structure of the CAPIF_API_Invoker_Management_API

Table 8.4.2.1-1 provides an overview of the resources and applicable HTTP methods.

Table 8.4.2.1-1: Resources and methods overview

Resource name	Resource URI	HTTP method or custom operation	Description
On-boarded API Invokers	{apiRoot} /api-invoker-management/v1 /onboardedInvokers	POST	On-boards a new API invoker by creating a API invoker profile
Individual On-boarded API Invoker	{apiRoot} /api-invoker-management/v1 /onboardedInvokers/{onboardingId}	DELETE	Off-boards an individual API invoker by deleting the API invoker profile identified by {onboardingId}

8.4.2.2 Resource: On-boarded API invokers

8.4.2.2.1 Description

The On-boarded API Invokers resource represents all the API invokers that are on-boarded at a given CAPIF core function.

8.4.2.2.2 Resource Definition

Resource URI: **{apiRoot}/api-invoker-management/v1/onboardedInvokers**

This resource shall support the resource URI variables defined in table 8.4.2.2.2-1.

Table 8.4.2.2.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See subclause 7.5

8.4.2.2.3 Resource Standard Methods

8.4.2.2.3.1 POST

This method shall support the URI query parameters specified in table 8.4.2.2.3.1-1.

Table 8.4.2.2.3.1-1: URI query parameters supported by the POST method on this resource

Name	Data type	P	Cardinality	Description
n/a				

This method shall support the request data structures specified in table 8.4.2.2.3.1-2 and the response data structures and response codes specified in table 8.4.2.2.3.1-3.

Table 8.4.2.2.3.1-2: Data structures supported by the POST Request Body on this resource

Data type	P	Cardinality	Description
APIInvokerEnrolmentDetails	M	1	Enrolment details of the API invoker including notification destination URI for any on-boarding related notifications and an optional list of APIs the API invoker intends to invoke while on-board.

Table 8.4.2.2.3.1-3: Data structures supported by the POST Response Body on this resource

Data type	P	Cardinality	Response codes	Description
APIInvokerEnrolmentDetails	M	1	201 Created	API invoker on-boarded successfully The URI of the created resource shall be returned in the "Location" HTTP header. A list of APIs the API invoker is allowed to invoke while on-board may also be included.
n/a			202 Accepted	The CAPIF core has accepted the Onboarding request and is processing it. When processing is completed, the CAPIF core function will send a Notify_Onboarding_Completion notification to the requesting API Invoker. See subclause 8.4.3.2.

8.4.2.2.4 Resource Custom Operations

None.

8.4.2.3 Resource: Individual On-boarded API Invoker

8.4.2.3.1 Description

The Individual On-boarded API Invokers resource represents an individual API invoker that is on-boarded at a given CAPIF core function.

8.4.2.3.2 Resource Definition

Resource URI: {apiRoot}/api-invoker-management/v1/onboardedInvokers/{onboardingId}

This resource shall support the resource URI variables defined in table 8.4.2.3.2-1.

Table 8.4.2.3.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See subclause 7.5
onboardingId	String identifying an individual on-boarded API invoker resource

8.4.2.3.3 Resource Standard Methods

8.4.2.3.3.1 DELETE

This method shall support the URI query parameters specified in table 8.4.2.3.3.1-1.

Table 8.4.2.3.3.1-1: URI query parameters supported by the DELETE method on this resource

Name	Data type	P	Cardinality	Description
n/a				

This method shall support the response codes specified in table 8.4.2.3.3.1-2 and the response data structures and response codes specified in table 8.4.2.3.3.1-3.

Table 8.4.2.3.3.1-2: Data structures supported by the DELETE Request Body on this resource

Data type	P	Cardinality	Description
n/a			

Table 8.4.2.3.3.1-3: Data structures supported by the DELETE Response Body on this resource

Data type	P	Cardinality	Response codes	Description
n/a			204 No Content	The individual on-boarded API invoker matching the onboardingId is deleted

8.3.2.3.4 Resource Custom Operations

None.

8.4.3 Notifications

8.4.3.1 General

The delivery of notifications shall conform to subclause 7.6.

Table 8.4.3.1-1: Notifications overview

Notification	Resource URI	HTTP method or custom operation	Description (service operation)
Notify_Onboarding_Completion	{notificationDestination}	POST	Notify API invoker of on-boarding result

8.4.3.2 Notify_Onboarding_Completion

8.4.3.2.1 Description

Notify_Onboarding_Completion is used by the CAPIF core function to notify an API invoker of the on-boarding result.

8.4.3.2.2 Notification definition

The POST method shall be used for Notify_Onboarding_Completion and the URI shall be the one provided by the API invoker during the on-boarding request.

Resource URI: **{notificationDestination}**

This method shall support the URI query parameters specified in table 8.4.3.2.2-1.

Table 8.4.3.2.2-1: URI query parameters supported by the POST method on this resource

Name	Data type	P	Cardinality	Description
n/a				

This method shall support the request data structures specified in table 8.4.3.2.2-2 and the response data structures and response codes specified in table 8.4.3.2.2-3.

Table 8.4.3.2.2-2: Data structures supported by the POST Request Body on this resource

Data type	P	Cardinality	Description
OnboardingNotification	M	1	Notification with on-boarding result

Table 8.4.3.2.2-3: Data structures supported by the POST Response Body on this resource

Data type	P	Cardinality	Response codes	Description
n/a			204 No Content	The receipt of the Notification is acknowledged.

8.4.4 Data Model

8.4.4.1 General

This subclause specifies the application data model supported by the API. Data types listed in subclause 7.2 apply to this API.

Table 8.4.4.1-1 specifies the data types defined specifically for the CAPIF_API_Invoker_Management_API service.

Table 8.4.4.1-1: CAPIF_API_Invoker_Management_API specific Data Types

Data type	Section defined	Description	Applicability
APIInvokerEnrolmentDetails	8.4.4.2.2	API invoker's enrolment details	
OnboardingNotificationDestination	8.4.4.2.3	Notification destination details.	
APIList	8.4.4.2.4	List of APIs	
OnboardingInformation	8.4.4.2.5	On-boarding information of the API invoker	
OnboardingRequestAck	8.4.4.2.6	Acknowledgement to received request.	
OnboardingNotification	8.4.4.2.7	Notification with on-boarding result	

Table 8.4.4.1-2 specifies data types re-used by the CAPIF_API_Invoker_Management_API service.

Table 8.4.4.1-2: Re-used Data Types

Data type	Reference	Comments	Applicability
ServiceAPIDescription	Subclause 8.1.4.2.2		
Uri	3GPP TS 29.122 [14]		
TestNotification	3GPP TS 29.122 [14]		
WebsockNotifConfig	3GPP TS 29.122 [14]		

8.4.4.2 Structured data types

8.4.4.2.1 Introduction

8.4.4.2.2 Type: APIInvokerEnrolmentDetails

Table 8.4.4.2.2-1: Definition of type APIInvokerEnrolmentDetails

Attribute name	Data type	P	Cardinality	Description	Applicability
apiInvokerId	string	O	0..1	API invoker ID assigned by the CAPIF core function to the API invoker while on-boarding the API invoker. Shall not be present in the HTTP POST request from the API invoker to the CAPIF core function, to on-board itself. Shall be present in all other HTTP requests and responses.	
onboardingInformation	OnboardingInformation	M	1	On-boarding information about the API invoker necessary for the CAPIF core function to on-board the API invoker.	
onboardingNotificationDestination	OnboardingNotificationDestination	M	1	Onboarding notification destination information provided by the API invoker.	
apiList	APIList	O	0..1	A list of APIs. When included by the API invoker in the HTTP request message, it lists the APIs that the API invoker intends to invoke while onboard. When included by the CAPIF core function in the HTTP response message, it lists the APIs that the API invoker is allowed to invoke while onboard.	
apiInvokerInformation	string	O	0..1	Generic information related to the API invoker such as details of the device or the application.	

8.4.4.2.3 Type: OnboardingNotificationDestination

Table 8.4.4.2.3-1: Definition of type OnboardingNotificationDestination

Attribute name	Data type	P	Cardinality	Description	Applicability
notificationDestination	Uri	M	1	URI where the notification should be delivered to.	
requestTestNotification	boolean	O	0..1	Set to true by Subscriber to request the CAPIF core function to send a test notification as defined in in subclause 7.6. Set to false or omitted otherwise.	Notification_test_event
websockNotifConfig	WebsockNotifConfig	O	0..1	Configuration parameters to set up notification delivery over Websocket protocol as defined in subclause 7.6.	Notification_websocket

8.4.4.2.4 Type: APIList

Table 8.4.4.2.4-1: Definition of type APIList

Attribute name	Data type	P	Cardinality	Description	Applicability
serviceAPIDescriptions	array(ServiceAPIDescription)	M	1..N	Definition of the service API	

8.4.4.2.5 Type: OnboardingInformation

Table 8.4.4.2.5-1: Definition of type OnboardingInformation

Attribute name	Data type	P	Cardinality	Description	Applicability
apiInvokerPublicKey	string	M	1	Public Key of API Invoker	
apiInvokerCertificate	string	O	0..1	API invoker's generic client certificate	

8.4.4.2.6 Type: OnboardingRequestAck

Table 8.4.4.2.6-1: Definition of type OnboardingRequestAck

Attribute name	Data type	P	Cardinality	Description	Applicability
onboardingNotificationDestination	OnboardingNotificationDestination	M	1	On-boarding notification destination related details	

8.4.4.2.7 Type: OnboardingNotification

Table 8.4.4.2.7-1: Definition of type OnboardingNotification

Attribute name	Data type	P	Cardinality	Description	Applicability
result	boolean	M	1	Set to "true" indicate successful on-boarding. Otherwise set to "false"	
resourceLocation	Uri	C	1	URI pointing to the new CAPIF resource created as a result of successful on-boarding. This attribute shall be present if 'result' attribute is set to "true". Otherwise it shall not be present.	
apiInvokerEnrolmentDetails	APIInvokerEnrolmentDetails	C	1	Enrolment details of the API invoker which are verified by the CAPIF administrator or API management. This attribute shall be present if 'result' attribute is set to "true". Otherwise it shall not be present.	
apiList	APIList	O	0..1	List of APIs API invoker is allowed to access. This attribute may be present if 'result' attribute is set to "true". Otherwise it shall not be present.	

8.4.4.3 Simple data types and enumerations

None.

8.4.5 Error Handling

General error responses are defined in subclause 7.7.

8.4.6 Feature negotiation

General feature negotiation procedures are defined in subclause 7.8. Table 8.4.6-1 lists the supported features for CAPIF_API_Invoker_Management_API.

Table 8.4.6-1: Supported Features

Feature number	Feature Name	Description
1	Notification_test_event	Testing of notification connection is supported according to subclause 7.6.
2	Notification_websocket	The delivery of notifications over Websocket is supported according to subclause 7.6. This feature requires that the Notification_test_event feature is also supported.

8.5 CAPIF_Security_API

8.5.1 API URI

The request URI used in each HTTP request from the API invoker or the API exposing function towards the CAPIF core function shall have the structure as defined in subclause 7.5 with the following clarifications:

- The {apiName} shall be "capif-security".
- The {apiVersion} shall be "v1".
- The {apiSpecificSuffixes} shall be set as described in subclause 8.5.2.

8.5.2 Resources

8.5.2.1 Overview

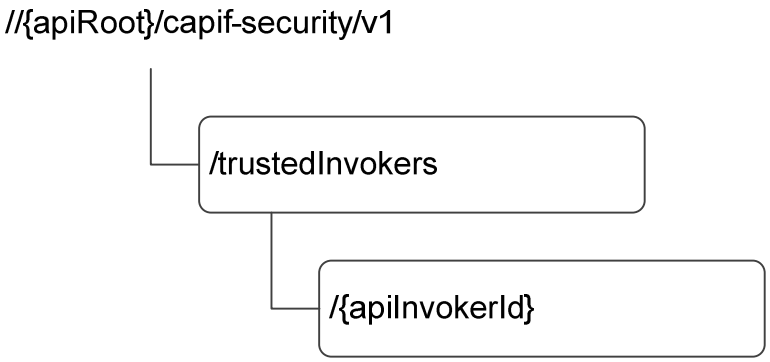

Figure 8.5.2.1-1: Resource URI structure of the CAPIF_Security_API

Table 8.5.2.1-1 provides an overview of the resources and applicable HTTP methods.

Table 8.5.2.1-1: Resources and methods overview

Resource name	Resource URI	HTTP method or custom operation	Description
Trusted API invokers	{apiRoot} /capif-security/v1 /trustedInvokers	POST	All trusted API invokers
Individual trusted API invoker	{apiRoot} /capif-security/v1 /trustedInvokers/{apiInvokerId}	GET	Retrieve authentication information of an API invoker
		DELETE	Revoke the authorization of the API invoker

8.5.2.2 Resource: Trusted API invokers

8.5.2.2.1 Description

The Trusted API Invokers resource represents all the API invokers that are trusted by the CAPIF core function and have received authentication information from the CAPIF core function.

8.5.2.2.2 Resource Definition

Resource URI: **{apiRoot}/capif-security/v1/trustedInvokers**

This resource shall support the resource URI variables defined in table 8.5.2.2.2-1.

Table 8.5.2.2.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See subclause 7.5

8.5.2.2.3 Resource Standard Methods

8.5.2.2.3.1 POST

This method shall support the URI query parameters specified in table 8.5.2.2.3.1-1.

Table 8.5.2.2.3.1-1: URI query parameters supported by the POST method on this resource

Name	Data type	P	Cardinality	Description
n/a				

This method shall support the request data structures specified in table 8.5.2.2.3.1-2 and the response data structures and response codes specified in table 8.5.2.2.3.1-3.

Table 8.5.2.2.3.1-2: Data structures supported by the POST Request Body on this resource

Data type	P	Cardinality	Description
ServiceSecurity	M	1	Security method request from the API invoker to the CAPIF core function. The request indicates a list of service APIs and a preferred method of security for the service APIs. The request also includes a notification destination URI for security related notifications.

Table 8.5.2.2.3.1-3: Data structures supported by the POST Response Body on this resource

Data type	P	Cardinality	Response codes	Description
ServiceSecurity	M	1	201 Created	Security method from the CAPIF core function to the API invoker is based on the received request. The response indicates the security method to be used for the service APIs The URI of the created resource shall be returned in the "Location" HTTP header.

8.5.2.2.4 Resource Custom Operations

None.

8.5.2.3 Resource: Individual trusted API invokers

8.5.2.3.1 Description

The Individual trusted API Invokers resource represents an individual API invokers that is trusted by the CAPIF core function and have received security related information from the CAPIF core function.

8.5.2.3.2 Resource Definition

Resource URI: **{apiRoot}/capif-security/v1/trustedInvokers/{apiInvokerId}**

This resource shall support the resource URI variables defined in table 8.5.2.3.2-1.

Table 8.5.2.3.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See subclause 7.5
apiInvokerId	String identifying an individual API invoker

8.5.2.3.3 Resource Standard Methods

8.5.2.3.3.1 GET

This method shall support the URI query parameters specified in table 8.5.2.3.3.1-1.

Table 8.5.2.3.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	P	Cardinality	Description
authenticationInfo	boolean	O	0..1	When set to 'true', it indicates the CAPIF core function to send the authentication information of the API invoker. Set to false or omitted otherwise.
authorizationInfo	boolean	O	0..1	When set to 'true', it indicates the CAPIF core function to send the authorization information of the API invoker. Set to false or omitted otherwise.

This method shall support the request data structures specified in table 8.5.2.3.3.1-2 and the response data structures and response codes specified in table 8.5.2.3.3.1-3.

Table 8.5.2.3.3.1-2: Data structures supported by the GET Request Body on this resource

Data type	P	Cardinality	Description
n/a			

Table 8.5.2.3.3.1-3: Data structures supported by the GET Response Body on this resource

Data type	P	Cardinality	Response codes	Description
ServiceSecurity	M	1	200 OK	The security related information of the API Invoker based on the request from the API exposing function.

8.5.2.3.3.2 DELETE

This method shall support the URI query parameters specified in table 8.5.2.3.3.2-1.

Table 8.5.2.3.3.2-1: URI query parameters supported by the DELETE method on this resource

Name	Data type	P	Cardinality	Description
n/a				

This method shall support the request data structures specified in table 8.5.2.3.3.2-2 and the response data structures and response codes specified in table 8.5.2.3.3.2-3.

Table 8.5.2.3.3.2-2: Data structures supported by the DELETE Request Body on this resource

Data type	P	Cardinality	Description
n/a			

Table 8.5.2.3.3.2-3: Data structures supported by the DELETE Response Body on this resource

Data type	P	Cardinality	Response codes	Description
n/a			204 No Content	Authorization of the API invoker revoked, and a notification is sent to the API invoker as specified in subclause 8.5.3.2

8.5.2.3.4 Resource Custom Operations

None.

8.5.3 Notifications

8.5.3.1 General

The delivery of notifications shall conform to subclause 7.6.

Table 8.5.3.1-1: Notifications overview

Notification	Resource URI	HTTP method or custom operation	Description (service operation)
Authorization revoked notification	{notificationDestination}	POST	Notify API invoker that the authorization rights are revoked by the API exposing function.

8.5.3.2 Authorization revoked notification

8.5.3.2.1 Description

Authorization revoked notification is used by the CAPIF core function to notify an API invoker that the authorization rights are revoked by the API exposing function.

8.5.3.2.2 Notification definition

The POST method shall be used for Authorization revoked notification and the URI shall be the one provided by the API invoker during the Obtain_Security_Method service operation.

Resource URI: **{notificationDestination}**

This method shall support the URI query parameters specified in table 8.5.3.2.2-1.

Table 8.5.3.2.2-1: URI query parameters supported by the POST method on this resource

Name	Data type	P	Cardinality	Description
n/a				

This method shall support the request data structures specified in table 8.5.3.2.2-2 and the response data structures and response codes specified in table 8.5.3.2.2-3.

Table 8.5.3.2.2-2: Data structures supported by the POST Request Body on this resource

Data type	P	Cardinality	Description
SecurityNotification	M	1	Notification with information related to revoked authorization.

Table 8.5.3.2.2-3: Data structures supported by the POST Response Body on this resource

Data type	P	Cardinality	Response codes	Description
n/a			204 No Content	The receipt of the Notification is acknowledged.

8.5.4 Data Model

8.5.4.1 General

This subclause specifies the application data model supported by the API.

Table 8.5.4.1-1 specifies the data types defined for the CAPIF service based interface protocol.

Table 8.5.4.1-1: Specific Data Types

Data type	Section defined	Description	Applicability
ServiceSecurity	8.5.4.2.2	Details of the security method for each service API interface. When included by the API invoker, it shall indicate the preferred method of security. When included by the CAPIF core function, it shall indicate the security method to be used for the service API interface.	
SecurityMethod	8.5.4.2.3	Interface details and the security method	
SecurityNotificationDestination	8.5.4.2.4	Notification destination details	
SecurityNotification	8.5.4.2.5	Revoked authorization notification details	

Table 8.5.4.1-2 specifies data types re-used by the CAPIF_Security_API service based interface:

Table 8.5.4.1-2: Re-used Data Types

Data type	Reference	Comments	Applicability
InterfaceDescription	Subclause 8.2.4.2.3	Details of the interface	
Uri	3GPP TS 29.122 [14]		
TestNotification	3GPP TS 29.122 [14]	Following differences apply: - The SCEF is the CAPIF core function; and - The SCS/AS is the Subscribing functional entity.	
WebsockNotifConfig	3GPP TS 29.122 [14]	Following differences apply: - The SCEF is the CAPIF core function; and - The SCS/AS is the Subscribing functional entity.	

8.5.4.2 Structured data types

8.5.4.2.1 Introduction

8.5.4.2.2 Type: ServiceSecurity

Table 8.5.4.2.2-1: Definition of type ServiceSecurity

Attribute name	Data type	P	Cardinality	Description	Applicability
securityPreferences	array(SecurityMethod)	O	0..N	API invoker's preference of the security method for a service API interface. May be present in the HTTP POST request from the API invoker to the CAPIF core function to negotiate the security mechanism. Shall not be present in any other HTTP request or HTTP response.	
selectedSecurityMethods	array(SecurityMethod)	C	0..N	Security method selected by the CAPIF core function for each service API interface. Shall be present in the HTTP POST response from the CAPIF core function to the API invoker. Shall not be present in any other HTTP request or HTTP response.	
apiInvokerSecurityMethods	array(SecurityMethod)	O	0..N	Security method for each service API interface selected for the API invoker by the CAPIF core function. May be present only in the HTTP GET response from the CAPIF core function to the API exposing function. Shall not be present in any other HTTP request or HTTP response.	
securityNotificationDestination	SecurityNotificationDestination	M	1	Security notification destination information provided by the API invoker.	

8.5.4.2.3 Type: SecurityMethod

Table 8.5.4.2.3-1: Definition of type SecurityMethod

Attribute name	Data type	P	Cardinality	Description	Applicability
interfaceDetails	InterfaceDescription	M	1	Details of the interface	
securityMethod	string	M	1	Security method for the interface	
authenticationInfo	string	O	0..1	Authentication related information	
authorizationInfo	string	O	0..1	Authorization related information	

Editor's Note: The data models should be improved to accommodate changes or improvements done in SA3.

8.5.4.2.4 Type: SecurityNotificationDestination

Table 8.5.4.2.4-1: Definition of type SecurityNotificationDestination

Attribute name	Data type	P	Cardinality	Description	Applicability
notificationDestination	Uri	M	1	URI where the notification should be delivered to.	
requestTestNotification	boolean	O	0..1	Set to true by API invoker to request the CAPIF core function to send a test notification as defined in in subclause 7.6. Set to false or omitted otherwise.	Notification_test_event
websockNotifConfig	WebsockNotifConfig	O	0..1	Configuration parameters to set up notification delivery over Websocket protocol as defined in subclause 7.6.	Notification_websocket

8.5.4.2.5 Type: SecurityNotification

Table 8.5.4.2.5-1: Definition of type SecurityNotification

Attribute name	Data type	P	Cardinality	Description	Applicability
apiInvokerId	string	M	1	String identifying the API invoker assigned by the CAPIF core function	
apiId	string	M	1	Identifier of the service API	
cause	Cause	M	1	The cause for revoking the API invoker authorization to the service API	

8.5.4.3 Simple data types and enumerations

8.5.4.3.1 Introduction

This subclause defines simple data types and enumerations that can be referenced from data structures defined in the previous subclauses.

8.5.4.3.2 Simple data types

The simple data types defined in table 8.5.4.3.2-1 shall be supported.

Table 8.5.4.3.2-1: Simple data types

Type Name	Type Definition	Description	Applicability
n/a			

8.5.4.3.3 Enumeration: Cause

Table 8.5.4.3.3-1: Enumeration Cause

Enumeration value	Description	Applicability
OVERLIMIT_USAGE	The revocation of the authorization of the API invoker is due to the overlimit usage of the service API	
UNEXPECTED_REASON	The revocation of the authorization of the API invoker is due to unexpected reason.	

8.5.5 Error Handling

General error responses are defined in subclause 7.7.

8.5.6 Feature negotiation

General feature negotiation procedures are defined in subclause 7.8. Table 8.5.6-1 lists the supported features for CAPIF_Security_API.

Table 8.5.6-1: Supported Features

Feature number	Feature Name	Description
1	Notification_test_event	Testing of notification connection is supported according to subclause 7.6.
2	Notification_websocket	The delivery of notifications over Websocket is supported according to subclause 7.6. This feature requires that the Notification_test_event feature is also supported.

8.6 CAPIF_Access_Control_Policy_API

8.6.1 API URI

The request URI used in each HTTP request from the API exposing function towards the CAPIF core function shall have the structure as defined in subclause 7.5 with the following clarifications:

- The {apiName} shall be "access-control-policy".
- The {apiVersion} shall be "v1".
- The {apiSpecificSuffixes} shall be set as described in subclause 8.6.2.

8.6.2 Resources

8.6.2.1 Overview

This resource is created by the CAPIF administrator on the CAPIF core function.

NOTE: The details of the mechanisms used to create the Access Control Policy List resource on the CAPIF core function is out of the scope of the present document.

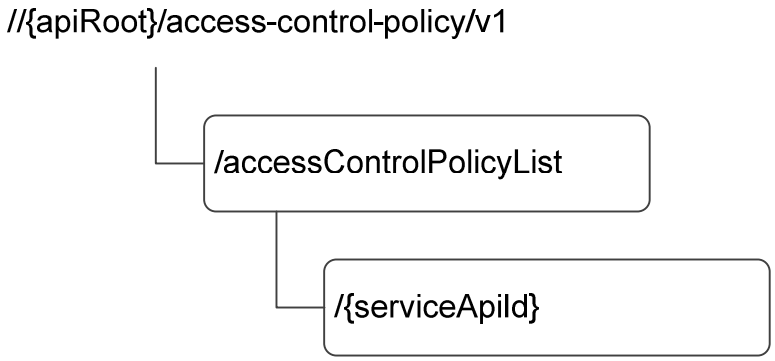

Figure 8.6.2.1-1: Resource URI structure of the CAPIF_Access_Control_Policy_API

Table 8.6.2.1-1 provides an overview of the resources and applicable HTTP methods.

Table 8.6.2.1-1: Resources and methods overview

Resource name	Resource URI	HTTP method or custom operation	Description
Access Control Policy List	{apiRoot} /access-control-policy/v1 /accessControlPolicyList/{serviceApiId}	GET	Retrieves the access control policy list for a published service API.

8.6.2.2 Resource: Access Control Policy List

8.6.2.2.1 Description

The Access Control Policy List resource represents the access control information for all the service APIs per API invoker.

8.6.2.2.2 Resource Definition

Resource URI: **{apiRoot}/access-control-policy/v1/accessControlPolicyList/{serviceApiId}**

This resource shall support the resource URI variables defined in table 8.6.2.2.2-1.

Table 8.6.2.2.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See subclause 7.5
serviceApiId	String identifying an individual published service API

8.6.2.2.3 Resource Standard Methods

8.6.2.2.3.1 GET

This method shall support the URI query parameters specified in table 8.6.2.2.3.1-1.

Table 8.6.2.2.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	P	Cardinality	Description
apiInvokerId	string	O	1	String identifying the API invoker

This method shall support the request data structures specified in table 8.6.2.2.3.1-2 and the response data structures and response codes specified in table 8.6.2.2.3.1-3.

Table 8.6.2.2.3.1-2: Data structures supported by the GET Request Body on this resource

Data type	P	Cardinality	Description
n/a			

Table 8.6.2.2.3.1-3: Data structures supported by the GET Response Body on this resource

Data type	P	Cardinality	Response codes	Description
accessControlPolicyList	M	1	200 OK	List of the access control policy applicable for the service API requested.

8.6.2.2.4 Resource Custom Operations

None.

8.6.3 Notifications

None.

8.6.4 Data Model

8.6.4.1 General

This subclause specifies the application data model supported by the API. Data types listed in subclause 7.2 apply to this API.

Table 8.6.4.1-1 specifies the data types defined specifically for the CAPIF_Access_Control_Policy_API service.

Table 8.6.4.1-1: CAPIF_Access_Control_Policy_API specific Data Types

Data type	Section defined	Description	Applicability
AccessControlPolicyList	8.6.4.2.2	Access control policy list	

Table 8.6.4.1-2 specifies data types re-used by the CAPIF_Access_Control_Policy_API service.

Table 8.6.4.1-2: Re-used Data Types

Data type	Reference	Comments	Applicability
n/a			

8.6.4.2 Structured data types

8.6.4.2.1 Introduction

This subclause defines data structures to be used in resource representations.

8.6.4.2.2 Type: AccessControlPolicyList

Table 8.6.4.2.2-1: Definition of type AccessControlPolicyList

Attribute name	Data type	P	Cardinality	Description	Applicability
apiInvokerPolicies	array(ApiInvokerPolicy)	M	0..N	Policy of each API invoker.	

8.6.4.2.3 Type: ApiInvokerPolicy

Table 8.6.4.2.3-1: Definition of type ApiInvokerPolicy

Attribute name	Data type	P	Cardinality	Description	Applicability
apiInvokerId	string	M	1	API invoker ID assigned by the CAPIF core function	
allowedTotalInvocations	integer	O	0..1	Total number of invocations allowed on the service API by the API invoker.	
allowedInvocationsPerSecond	integer	O	0..1	Invocations per second allowed on the service API by the API invoker.	
allowedInvocationTimeRangeList	array(TimeRangeList)	O	1..N	The time ranges during which the invocations are allowed on the service API by the API invoker.	

8.6.4.2.4 Type: TimeRangeList

Table 8.6.4.2.4-1: Definition of type TimeRangeList

Attribute name	Data type	P	Cardinality	Description	Applicability
startTime	DateTime	M	1	The start time for the invocations to be allowed on the service API by the API invoker.	
endTime	DateTime	M	1	The end time for the invocations to be allowed on the service API by the API invoker.	

8.6.4.3 Simple data types and enumerations

None.

8.6.5 Error Handling

General error responses are defined in subclause 7.7.

8.6.6 Feature negotiation

General feature negotiation procedures are defined in subclause 7.8.

Table 8.6.8-1: Supported Features

Feature number	Feature Name	Description
n/a		

8.7 CAPIF_Logging_API_Invocation_API

8.7.1 API URI

The request URI used in each HTTP request from the API exposing function towards the CAPIF core function shall have the structure as defined in subclause 7.5 with the following clarifications:

- The {apiName} shall be "api-invocation-logs".
- The {apiVersion} shall be "v1".
- The {apiSpecificSuffixes} shall be set as described in subclause 8.7.2

8.7.2 Resources

8.7.2.1 Overview

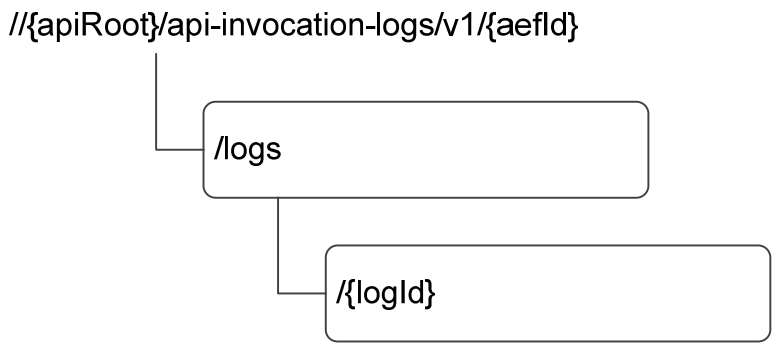

Figure 8.7.2.1-1: Resource URI structure of the CAPIF_Logging_API_Invocation_API

Table 8.7.2.1-1 provides an overview of the resources and applicable HTTP methods.

Table 8.7.2.1-1: Resources and methods overview

Resource name	Resource URI	HTTP method or custom operation	Description
Logs	{apiRoot} /api-invocation-logs/v1 /{aefId}/logs	POST	Creates a new log entry for service API invocations
Individual log	{apiRoot} /api-invocation-logs/v1 /{aefId}/logs/{logId}	n/a	Individual log entry

8.7.2.2 Resource: Logs

8.7.2.2.1 Description

The Logs resource represents all the log entries created by a API exposing function at CAPIF core function.

8.7.2.2.2 Resource Definition

Resource URI: **{apiRoot}/api-invocation-logs/v1/{aefId}/logs**

This resource shall support the resource URI variables defined in table 8.7.2.2.2-1.

Table 8.7.2.2.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See subclause 7.5
aefId	Identity of the API exposing function

8.7.2.2.3 Resource Standard Methods

8.7.2.2.3.1 POST

This method shall support the URI query parameters specified in table 8.7.2.2.3.1-1.

Table 8.7.2.2.3.1-1: URI query parameters supported by the POST method on this resource

Name	Data type	P	Cardinality	Description
n/a				

This method shall support the request data structures specified in table 8.7.2.2.3.1-2 and the response data structures and response codes specified in table 8.7.2.2.3.1-3.

Table 8.7.2.2.3.1-2: Data structures supported by the POST Request Body on this resource

Data type	P	Cardinality	Description
InvocationLogs	M	201 Created	Log of service API invocations provided by API exposing function to store on the CAPIF core function.

Table 8.7.2.2.3.1-3: Data structures supported by the POST Response Body on this resource

Data type	P	Cardinality	Response codes	Description
InvocationLogs	M	1	201 Created	Log of service API invocations provided by API exposing function successfully stored on the CAPIF core function. The URI of the created resource shall be returned in the "Location" HTTP header.

8.7.2.2.4 Resource Custom Operations

None.

8.7.3 Notifications

None.

8.7.4 Data Model

8.7.4.1 General

This subclause specifies the application data model supported by the API.

Table 8.7.4.1-1 specifies the data types defined for the CAPIF service based interface protocol.

Table 8.7.4.1-1: Specific Data Types

Data type	Section defined	Description	Applicability
InvocationLogs	8.7.4.2.2	Set of Service API invocation logs to be stored on CAPIF core function	
Log	8.7.4.2.3	Individual log entries	

Table 8.7.4.1-2 specifies data types re-used by the CAPIF_Logging_API_Invocation_API service:

Table 8.7.4.1-2: Re-used Data Types

Data type	Reference	Comments	Applicability
Ipv4Addr	3GPP TS 29.122 [14]		
Ipv6Addr	3GPP TS 29.122 [14]		
Port	3GPP TS 29.122 [14]		
InterfaceDescription	Subclause 8.2.4.2.3	Details of the interface	

8.7.4.2 Structured data types

8.7.4.2.1 Introduction

8.7.4.2.2 Type: InvocationLogs

Table 8.7.4.2.2-1: Definition of type InvocationLogs

Attribute name	Data type	P	Cardinality	Description	Applicability
aefId	string	M	1	Identity information of the API exposing function requesting logging of service API invocations	
apiInvokerId	string	M	1	Identity of the API invoker which invoked the service API	
ipv4Addr	Ipv4Addr	C	0..1	String identifying an IPv4 address of the API invoker. This attribute shall not be present if ipv6Addr attribute is present.	
ipv6Addr	Ipv6Addr	C	0..1	String identifying an IPv6 address of the API invoker. This attribute shall not be present if ipv4Addr attribute is present.	
port	Port	C	0..1	Port. This attribute shall be present if either ipv4Addr or the ipv6Addr attribute is present.	
logs	array(Log)	M	1..N	Service API invocation log	

8.7.4.2.3 Type: Log

Table 8.7.4.2.3-1: Definition of type Log

Attribute name	Data type	P	Cardinality	Description	Applicability
apiId	string	M	1	String identifying the API invoked.	
apiInvoked	string	M	1	Name of the API which was invoked	
version	number	M	1	Version of the API which was invoked	
resourceName	string	M	1	Name of the specific resource invoked	
operation	string	M	1	Operation that was invoked on the API	
result	string	M	1	Result or output of the invocation	
invocationTime	DateTime	O	0..1	Date on which it was invoked	
parameters	string	O	0..1	List of input parameters	
interfaceDescription	InterfaceDescription	O	0..1	Interface description of the API invoked.	

8.7.4.3 Simple data types and enumerations

None.

8.7.5 Error Handling

General error responses are defined in subclause 7.7.

8.7.6 Feature negotiation

Table 8.7.8-1: Supported Features

Feature number	Feature Name	Description
n/a		

8.8 CAPIF_Auditing_API

8.8.1 API URI

The request URI used in each HTTP request from the API management function towards the CAPIF core function shall have the structure as defined in subclause 7.5 with the following clarifications:

- The {apiName} shall be "logs".
- The {apiVersion} shall be "v1".
- The {apiSpecificSuffixes} shall be set as described in subclause 8.8.2.

8.8.2 Resources

8.8.2.1 Overview

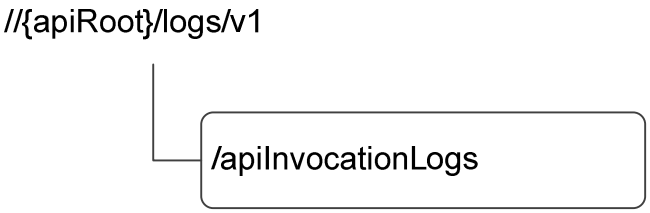

Figure 8.8.2.1-1: Resource URI structure of the CAPIF_Auditing_API

Table 8.8.2.1-1 provides an overview of the resources and applicable HTTP methods.

Table 8.8.2.1-1: Resources and methods overview

Resource name	Resource URI	HTTP method or custom operation	Description
All service API invocation logs (Store)	{apiRoot} /logs/v1/apiInvocationLogs	GET	Query and retrieve service API invocation logs stored on the CAPIF core function

8.8.2.2 Resource: All service API invocation logs

8.8.2.2.1 Description

The All service API invocation logs resource represents a collection of service API invocation logs stored on the CAPIF core function. The resource is modelled as a Store resource archetype (see annex C.3 of 3GPP TS 29.501 [18])

8.8.2.2.2 Resource Definition

Resource URI: **{apiRoot}/logs/v1/apiInvocationLogs**

This resource shall support the resource URI variables defined in table 8.8.2.2.2-1.

Table 8.8.2.2.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	See subclause 7.5

8.8.2.2.3 Resource Standard Methods

8.8.2.2.3.1 GET

This method shall support the URI query parameters specified in table 8.8.2.2.3.1-1.

Table 8.8.2.2.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	P	Cardinality	Description
aefId	string	O	0..1	String identifying the API exposing function
apiInvokerId	string	O	0..1	String identifying the API invoker which invoked the service API
ipv4Addr	Ipv4Addr	C	0..1	String identifying a IPv4 address of the API invoker. This attribute shall not be present if ipv6Addr attribute is present.
ipv6Addr	Ipv6Addr	C	0..1	String identifying a IPv6 address of the API invoker. This attribute shall not be present if ipv4Addr attribute is present.
port	Port	C	0..1	Port. This attribute shall be present if either ipv4Addr or the ipv6Addr attribute is present.
timeRangeStart	DateTime	O	0..1	Start time of the invocation time range
timeRangeEnd	DateTime	O	0..1	End time of the invocation time range
apiId	string	O	0..1	String identifying the API invoked.
apiName	String	O	0..1	Name of the API which was invoked
version	number	O	0..1	Version of the API which was invoked
operation	string	O	0..1	Operation that was invoked on the API
result	string	O	0..1	Result or output of the invocation
resourceName	string	O	0..1	Name of the specific resource invoked
interfaceDescription	InterfaceDescription	O	0..1	Interface description of the API invoked.

This method shall support the request data structures specified in table 8.8.2.2.3.1-2 and the response data structures and response codes specified in table 8.8.2.2.3.1-3.

Table 8.8.2.2.3.1-2: Data structures supported by the GET Request Body on this resource

Data type	P	Cardinality	Description
n/a			

Table 8.8.2.2.3.1-3: Data structures supported by the GET Response Body on this resource

Data type	P	Cardinality	Response codes	Description
array(InvocationLogs)	O	1..N	200 OK	Result of the query operation along with fetched service API invocation log data.
ProblemDetails	M	1	414 URI Too Long	Indicates that the server is refusing to service the request because the request-target is too long.

8.8.2.2.4 Resource Custom Operations

None.

8.8.3 Notifications

None.

8.8.4 Data Model

8.8.4.1 General

This subclause specifies the application data model supported by the API.

Table 8.8.4.1-1 specifies the data types defined for the CAPIF service based interface protocol.

Table 8.8.4.1-1: Specific Data Types

Data type	Section defined	Description	Applicability
n/a			

Table 8.8.4.1-2 specifies data types re-used by the CAPIF_Auditing_API service:

Table 8.8.4.1-2: Re-used Data Types

Data type	Reference	Comments	Applicability
InvocationLogs	Subclause 8.7.4.2.2	Logs of service API invocations stored on the CAPIF core function.	
ProblemDetails	3GPP TS 29.122 [14]		
Ipv4Addr	3GPP TS 29.122 [14]		
Ipv6Addr	3GPP TS 29.122 [14]		
Uri	3GPP TS 29.122 [14]		
Port	3GPP TS 29.122 [14]		

8.8.4.2 Structured data types

None.

8.8.4.3 Simple data types and enumerations

None.

8.8.5 Error Handling

General error responses are defined in subclause 7.7.

8.8.6 Feature negotiation

General feature negotiation procedures are defined in subclause 7.8.

Table 8.8.6-1: Supported Features

Feature number	Feature Name	Description
n/a		

9 AEF API Definition

9.1 AEF_Authentication_API

9.1.1 API URI

The request URI used in each HTTP request from the API invoker towards the API exposing function shall have the following structure:

- The {apiRoot} shall be as defined in the service API specification using CAPIF.
- The {apiName} shall be "auth-initiation".
- The {apiVersion} shall be "v1".
- The {apiSpecificSuffixes} shall be set as described in subclause 9.1.2.

9.1.2 Resources

9.1.2.1 Overview

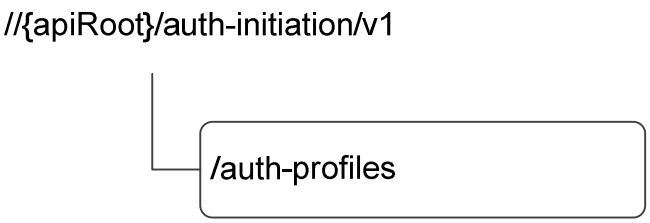

Figure 9.1.2.1-1: Resource URI structure of the AEF_Authentication_API

Table 9.1.2.1-1 provides an overview of the resources and applicable HTTP methods.

Table 9.1.2.1-1: Resources and methods overview

Resource name	Resource URI	HTTP method or custom operation	Description
API invoker authentication profiles	{apiRoot} /auth-initiation/v1 /auth-profiles	GET	Checks if the API exposing function has credentials for API invoker authentication

9.1.2.2 Resource: API invoker authentication profiles

9.1.2.2.1 Description

The API invoker authentication profiles represents all the authentication profiles of different API invokers available at the API exposing function.

9.1.2.2.2 Resource Definition

Resource URI: **{apiRoot}/auth-initiation /v1/auth-profiles**

This resource shall support the resource URI variables defined in table 9.1.2.2.2-1.

Table 9.1.2.2.2-1: Resource URI variables for this resource

Name	Definition
apiRoot	Defined in the service API specification using CAPIF

9.1.2.2.3 Resource Standard Methods

9.1.2.2.3.1 GET

This method shall support the URI query parameters specified in table 9.1.2.2.3.1-1.

Table 9.1.2.2.3.1-1: URI query parameters supported by the GET method on this resource

Name	Data type	P	Cardinality	Description
apiInvokerId	string	M	1	API invoker ID assigned by the CAPIF core function to the API invoker while on-boarding the API invoker.

This method shall support the request data structures specified in table 9.1.2.2.3.1-2 and the response data structures and response codes specified in table 9.1.2.2.3.1-3.

Table 9.1.2.2.3.1-2: Data structures supported by the GET Request Body on this resource

Data type	P	Cardinality	Description
n/a			

Table 9.1.2.2.3.1-3: Data structures supported by the POST Response Body on this resource

Data type	P	Cardinality	Response codes	Description
n/a			204 No Content	API exposing function confirms that the authentication profile is available and it is ready for authentication of the API invoker.

9.1.2.2.4 Resource Custom Operations

None.

9.1.3 Notifications

None.

9.1.4 Data Model

9.1.4.1 General

This subclause specifies the application data model supported by the API. Data types listed in subclause 7.2 apply to this API.

Table 9.1.4.1-1 specifies the data types defined specifically for the AEF_Authentication_API service.

Table 9.1.4.1-1: AEF_Authentication_API specific Data Types

Data type	Section defined	Description	Applicability
n/a			

Table 9.1.4.1-2 specifies data types re-used by the CAPIF_API_Invoker_Management_API service.

Table 9.1.4.1-2: Re-used Data Types

Data type	Reference	Comments	Applicability
n/a			

9.1.4.2 Structured data types

None.

9.1.4.3 Simple data types and enumerations

None.

9.1.5 Error Handling

General error responses are defined in the service API specification using CAPIF.

9.1.6 Feature negotiation

General feature negotiation procedures are defined in the service API specification using CAPIF.

Table 9.1.6-1: Supported Features

Feature number	Feature Name	Description
n/a		

10 Security

10.1 General

Security methods for CAPIF are specified in 3GPP TS 33.122 [16].

10.1 CAPIF-1/1e security

Secure communication between API invoker and CAPIF core function over CAPIF-1 or CAPIF-1e reference point, using a TLS protocol based connection is defined in 3GPP TS 33.122 [16].

For Onboard_API_Invoker service operation of the CAPIF_API_Invoker_Management_API, the TLS protocol based connection shall be established using server certificate as defined in 3GPP TS 33.122 [16].

For rest of the CAPIF APIs, the TLS protocol based connection shall be established with certificate based mutual authentication as defined in 3GPP TS 33.122 [16].

10.2 CAPIF-2/2e security and securely invoking service APIs

For secure communication between API invoker and API exposing function and ensuring secure invocations of service APIs, the API invoker:

- shall negotiate the security method with the CAPIF core function using the Obtain_Security_Method service operation of the CAPIF_Security_API;
- shall initiate the authentication with the API exposing function using the Authentication_Initiation_Request service operation of the AEF_Authentication _API; and
- shall establish a secure connection with the API exposing function as defined in 3GPP TS 33.122 [16], using the method negotiated with the CAPIF core function.

Annex A (normative): OpenAPI specification

A.1 General

This subclause will describe the purpose of the Annex.

A.2 CAPIF_Discover_Service_API

```
openapi: 3.0.0
info:
  description: Discover_Service API
  title: CAPIF_Discover_Service_API
  version: "1.PreR15.0.0"servers:
  - url: '{apiRoot}/service-apis/v1'
    variables:
      apiRoot:
        default: https://demohost.com
        description: apiRoot as defined in subclause 7.5 of 3GPP TS 29.222.
paths:
  /allServiceAPIs:
    get:
      description: Discover published service APIs and retrieve a collection of APIs according to
certain filter criteria.
      parameters:
        - name: apiInvokerId
          in: query
          description: String identifying the API invoker assigned by the CAPIF core function.
          required: true
          schema:
            type: string
        - name: serviceName
          in: query
          description: Name of the service.
          schema:
            type: string
        - name: apiName
          in: query
          description: API name.
          schema:
            type: string
        - name: apiVersion
          in: query
          description: API version.
          schema:
            type: string
        - name: commType
          in: query
          description: Communication type used by the API (e.g. request/response or
subscribe/notify).
          schema:
            $ref: 'TS29222_CAPIF_Publish_Service_API.yaml#/components/schemas/CommunicationType'
        - name: interfaceDescription
          in: query
          description: Interface details (e.g. domain name, ipv4/6Addr, port).
          schema:
            type: array
            items:
              $ref:
'TS29222_CAPIF_Publish_Service_API.yaml#/components/schemas/InterfaceDescription'
        - name: dataFormat
          in: query
          description: Data formats used by the API (e.g. serialization protocol JSON used).
          schema:
            $ref: 'TS29222_CAPIF_Publish_Service_API.yaml#/components/schemas/DataFormat'
      responses:
        '200':
          description: The response body contains the result of the search over the list of
registered APIs.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/DiscoveredAPIs'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
```

```
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '414':
          description: URI Too Long
          content:
            application/problem+json:
              schema:
                $ref: 'TS29122_CommonData.yaml#/components/schemas/ProblemDetails'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
components:
  schemas:
    DiscoveredAPIs:
      type: array
      properties:
        apiId:
          type: string
          description: Identifier of the service API
        serviceAPIDescription:
          $ref: 'TS29222_CAPIF_Publish_Service_API.yaml#/components/schemas/ServiceAPIDescription'
      required:
        - apiId
        - serviceAPIDescription
```

A.3 CAPIF_Publish_Service_API

```
openapi: 3.0.0
info:
  title: CAPIF_Publish_Service_API
  version: "1.PreR15.0.0"
servers:
  - url: '{apiRoot}/published-apis/v1'
    variables:
      apiRoot:
        default: https://demohost.com
        description: apiRoot as defined in subclause 7.5 of 3GPP TS 29.222.
paths:

# APF published API

  /{apfId}/service-apis:
    post:
      description: Publish a new API.
      parameters:
        - name: apfId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/apfId'
      requestBody:
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/ServiceAPIDescription'
      responses:
        '201':
          description: Service API published successfully The URI of the created resource shall be
returned in the "Location" HTTP header.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/ServiceAPIDescription'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
```

```
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
    get:
      description: Retrieve all published APIs.
      parameters:
        - name: apfId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/apfId'
      responses:
        '200':
          description: Definition of all service API(s) published by the API publishing function.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/ServiceAPIDescription'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'

# Individual APF published API

  /{apfId}/service-apis/{serviceApiId}:
    get:
      description: Retrieve a published service API.
      parameters:
        - name: serviceApiId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/serviceApiId'
        - name: apfId
          in: path
          required: true
          schema:
            $ref: '#/components/schemas/apfId'
      responses:
        '200':
          description: Definition of all service API published by the API publishing function.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/ServiceAPIDescription'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
```

```
        $ref: 'TS29122_CommonData.yaml#/components/responses/500'
      '503':
        $ref: 'TS29122_CommonData.yaml#/components/responses/503'
      default:
        $ref: 'TS29122_CommonData.yaml#/components/responses/default'
  put:
    description: Update a published service API.
    parameters:
      - name: serviceApiId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/serviceApiId'
      - name: apfId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/apfId'
    requestBody:
      required: true
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/ServiceAPIDescription'
    responses:
      '200':
        description: Definition of service API updated successfully.
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/ServiceAPIDescription'
      '400':
        $ref: 'TS29122_CommonData.yaml#/components/responses/400'
      '401':
        $ref: 'TS29122_CommonData.yaml#/components/responses/401'
      '403':
        $ref: 'TS29122_CommonData.yaml#/components/responses/403'
      '404':
        $ref: 'TS29122_CommonData.yaml#/components/responses/404'
      '409':
        $ref: 'TS29122_CommonData.yaml#/components/responses/409'
      '412':
        $ref: 'TS29122_CommonData.yaml#/components/responses/412'
      '500':
        $ref: 'TS29122_CommonData.yaml#/components/responses/500'
      '503':
        $ref: 'TS29122_CommonData.yaml#/components/responses/503'
      default:
        $ref: 'TS29122_CommonData.yaml#/components/responses/default'
  delete:
    description: Unpublish a published service API.
    parameters:
      - name: serviceApiId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/serviceApiId'
      - name: apfId
        in: path
        required: true
        schema:
          $ref: '#/components/schemas/apfId'
    responses:
      '204':
        description: The individual published service API matching the serviceAPiId is deleted.
      '400':
        $ref: 'TS29122_CommonData.yaml#/components/responses/400'
      '401':
        $ref: 'TS29122_CommonData.yaml#/components/responses/401'
      '403':
        $ref: 'TS29122_CommonData.yaml#/components/responses/403'
      '404':
        $ref: 'TS29122_CommonData.yaml#/components/responses/404'
      '412':
        $ref: 'TS29122_CommonData.yaml#/components/responses/412'
      '500':
        $ref: 'TS29122_CommonData.yaml#/components/responses/500'
      '503':
```

```
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'

# Components

components:
  schemas:
# Data types uses as path variables
    apfId:
      type: string
      description: Identification of the API publishing function.
    serviceApiId:
      type: string
      description: String identifying an individual published service API.
# Data Type for representations
    ServiceAPIDescription:
      type: object
      properties:
        apiName:
          type: string
          description: API name
        apiID:
          type: string
          description: API identifier assigned by the CAPIF core function to the published service
API. Shall not be present in the HTTP POST request from the API publishing function to the CAPIF
core function. Shall be present in the HTTP POST response from the CAPIF core function to the API
publishing function.
        apiVersion:
          type: string
          description: API version
        serviceName:
          type: string
          description: Name of the service to which the API belongs
        interfaceDescriptions:
          type: array
          items:
            $ref: '#/components/schemas/InterfaceDescription'
          minItems: 0
          description: Interface details
        commType:
          $ref: '#/components/schemas/CommunicationType'
        dataFormat:
          $ref: '#/components/schemas/DataFormat'
        description:
          type: string
          description: Text description of the API
        uris:
          type: array
          items:
            $ref: 'TS29122_CommonData.yaml#/components/schemas/Uri'
          minItems: 0
          description: Relative URI (s) of the API
        domainName:
          type: string
          description: Domain to which API belongs to
        protocol:
          $ref: '#/components/schemas/Protocol'
      required:
        - apiName
    InterfaceDescription:
      type: object
      properties:
        ipv4Addr:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Ipv4Addr'
        ipv6Addr:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Ipv6Addr'
        port:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Port'
        securityMethods:
          type: array
          items:
            $ref: '#/components/schemas/SecurityMethods'
          minItems: 1
          description: Security methods supported by the interface
    Protocol:
      anyOf:
      - type: string
```

```
        enum:
          - HTTP_1_1
          - HTTP_2
      - type: string
        description: >
          This string provides forward-compatibility with future
          extensions to the enumeration but is not used to encode
          content defined in the present version of this API.
      description: >
        Possible values are
        - HTTP_1_1: HTTP version 1.1
        - HTTP_2: HTTP version 2
    CommunicationType:
      anyOf:
      - type: string
        enum:
          - REQUEST_RESPONSE
          - SUBSCRIBE_NOTIFY
      - type: string
        description: >
          This string provides forward-compatibility with future
          extensions to the enumeration but is not used to encode
          content defined in the present version of this API.
      description: >
        Possible values are
        - REQUEST_RESPONSE: The communication is of the type request-response
        - SUBSCRIBE_NOTIFY: The communication is of the type subscribe-notify
    DataFormat:
      anyOf:
      - type: string
        enum:
          - JSON
      - type: string
        description: >
          This string provides forward-compatibility with future
          extensions to the enumeration but is not used to encode
          content defined in the present version of this API.
      description: >
        Possible values are
        - JSON: JavaScript Object Notation
    SecurityMethods:
      anyOf:
      - type: string
        enum:
          - PSK
          - PKI
          - OAUTH
      - type: string
        description: >
          This string provides forward-compatibility with future
          extensions to the enumeration but is not used to encode
          content defined in the present version of this API.
      description: >
        Possible values are
        - PSK: Security method 1 (Using TLS-PSK) as described in 3GPP TS 33.122
        - PKI: Security method 2 (Using PKI) as described in 3GPP TS 33.122
        - OAUTH: Security method 3 (TLS with OAuth token) as described in 3GPP TS 33.122
```

A.4 CAPIF_Events_API

```
openapi: 3.0.0
info:
  title: CAPIF_Events_API
  version: "1.PreR15.0.0"
servers:
  - url: '{apiRoot}/capif-events/v1'
    variables:
      apiRoot:
        default: https://demohost.com
        description: apiRoot as defined in subclause 7.5 of 3GPP TS 29.222

paths:
  /{subscriberId}/subscriptions:
    post:
      description: Creates a new individual CAPIF Event Subscription.
      parameters:
        - name: subscriberId
```

```
        in: path
        description: Identifier of the Subscriber
        required: true
        schema:
          type: string
    requestBody:
      required: true
      content:
        application/json:
          schema:
            $ref: '#/components/schemas/EventSubscription'
    callbacks:
      notificationDestination:
        '{request.body#/notificationDestination}':
          post:
            requestBody:  # contents of the callback message
              required: true
              content:
                application/json:
                  schema:
                    $ref: '#/components/schemas/EventNotification'
            responses:
              '204':
                description: No Content (successful notification)
              '400':
                $ref: 'TS29122_CommonData.yaml#/components/responses/400'
              '401':
                $ref: 'TS29122_CommonData.yaml#/components/responses/401'
              '403':
                $ref: 'TS29122_CommonData.yaml#/components/responses/403'
              '409':
                $ref: 'TS29122_CommonData.yaml#/components/responses/409'
              '411':
                $ref: 'TS29122_CommonData.yaml#/components/responses/411'
              '412':
                $ref: 'TS29122_CommonData.yaml#/components/responses/412'
              '413':
                $ref: 'TS29122_CommonData.yaml#/components/responses/413'
              '500':
                $ref: 'TS29122_CommonData.yaml#/components/responses/500'
              '503':
                $ref: 'TS29122_CommonData.yaml#/components/responses/503'
              default:
                $ref: 'TS29122_CommonData.yaml#/components/responses/default'
    responses:
      '201':
        description: Created (Successful creation of subscription)
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/EventSubscription'
      '400':
        $ref: 'TS29122_CommonData.yaml#/components/responses/400'
      '401':
        $ref: 'TS29122_CommonData.yaml#/components/responses/401'
      '403':
        $ref: 'TS29122_CommonData.yaml#/components/responses/403'
      '409':
        $ref: 'TS29122_CommonData.yaml#/components/responses/409'
      '411':
        $ref: 'TS29122_CommonData.yaml#/components/responses/411'
      '412':
        $ref: 'TS29122_CommonData.yaml#/components/responses/412'
      '413':
        $ref: 'TS29122_CommonData.yaml#/components/responses/413'
      '500':
        $ref: 'TS29122_CommonData.yaml#/components/responses/500'
      '503':
        $ref: 'TS29122_CommonData.yaml#/components/responses/503'
      default:
        $ref: 'TS29122_CommonData.yaml#/components/responses/default'

  /{subscriberId}/subscriptions/{subscriptionId}:
    delete:
      description: Deletes an individual CAPIF Event Subscription.
      parameters:
        - name: subscriberId
          in: path
```

```
          description: Identifier of the Subscriber
          required: true
          schema:
            type: string
        - name: subscriptionId
          in: path
          description: Identifier of an individual Events Subscription
          required: true
          schema:
            type: string
      responses:
        '204':
          description: The individual CAPIF Events Subscription matching the subscriptionId is
deleted.
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'

components:
  schemas:
    EventSubscription:
      type: object
      properties:
        events:
          type: array
          items:
            $ref: '#/components/schemas/CAPIFEvent'
          minItems: 1
          description: Subscribed events
        notificationDestination:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Uri'
        requestTestNotification:
          type: boolean
          description: Set to true by Subscriber to request the CAPIF core function to send a test
notification as defined in in subclause 7.6. Set to false or omitted otherwise.
        websockNotifConfig:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/WebsockNotifConfig'
      required:
        - events
        - notificationDestination
    EventNotification:
      type: object
      properties:
        subscriptionId:
          type: string
          description: Identifier of the subscription resource to which the notification is related
– CAPIF resource identifier
        events:
          $ref: '#/components/schemas/CAPIFEvent'
      required:
        - subscriptionId
        - events
    CAPIFEvent:
      anyOf:
      - type: string
        enum:
          - SERVICE_API_AVAILABLE
          - SERVICE_API_UNAVAILABLE
          - SERVICE_API_UPDATE
          - API_INVOKER_ONBOARDED
          - API_INVOKER_OFFBOARDED
          - SERVICE_API_INVOCATION_SUCCESS
          - SERVICE_API_INVOCATION_FAILURE
          - ACCESS_CONTROL_POLICY_UPDATE
          - ACCESS_CONTROL_POLICY_UNAVAILABLE
```

```
          - API_INVOKER_AUTHORIZATION_REVOKED
      - type: string
        description: >
          This string provides forward-compatibility with future
          extensions to the enumeration but is not used to encode
          content defined in the present version of this API.
      description: >
        Possible values are
        - SERVICE_API_AVAILABLE: Events related to the availability of service APIs after the
service APIs are published.
        - SERVICE_API_UNAVAILABLE: Events related to the unavailability of service APIs after the
service APIs are unpublished.
        - SERVICE_API_UPDATE: Events related to change in service API information.
        - API_INVOKER_ONBOARDED: Events related to API invoker onboarded to CAPIF.
        - API_INVOKER_OFFBOARDED: Events related to API invoker offboarded from CAPIF.
        - SERVICE_API_INVOCATION_SUCCESS: Events related to the successful invocation of service
APIs.
        - SERVICE_API_INVOCATION_FAILURE: Events related to the failed invocation of service APIs.
        - ACCESS_CONTROL_POLICY_UPDATE: Events related to the update for the access control policy
related to the service APIs.        - ACCESS_CONTROL_POLICY_UNAVAILABLE: Events related to the
unavailability of the access control policy related to the service APIs.
        - API_INVOKER_AUTHORIZATION_REVOKED: Events related to the revocation of the authorization
of API invokers to access the service APIs.
```

A.5 CAPIF_API_Invoker_Management_API

```
openapi: 3.0.0
info:
  title: CAPIF_API_Invoker_Management_API
  version: "1.PreR15.0.0"
servers:
  - url: '{apiRoot}/api-invoker-management/v1'
    variables:
      apiRoot:
        default: https://demohost.com
        description: apiRoot as defined in subclause 7.5 of 3GPP TS 29.222

paths:
  /onboardedInvokers:
    post:
      description: Creates a new individual API Invoker profile.
      requestBody:
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/APIInvokerEnrolmentDetails'
      callbacks:
        notificationDestination:
          '{request.body#/notificationDestination}':
            post:
              description: Notify the API Invoker about the onboarding completion
              requestBody:  # contents of the callback message
                required: true
                content:
                  application/json:
                    schema:
                      $ref: '#/components/schemas/OnboardingNotification'
              responses:
                '204':
                  description: No Content (successful onboarding notification)
                '400':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/400'
                '401':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/401'
                '403':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/403'
                '409':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/409'
                '411':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/411'
                '412':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/412'
                '413':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/413'
                '500':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/500'
```

```
                '503':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/503'
                default:
                  $ref: 'TS29122_CommonData.yaml#/components/responses/default'
      responses:
        '201':
          description: API invoker on-boarded successfully
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/APIInvokerEnrolmentDetails'
        '202':
          description: The CAPIF core has accepted the Onboarding request and is processing it.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/OnboardingRequestAck'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '411':
          $ref: 'TS29122_CommonData.yaml#/components/responses/411'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '413':
          $ref: 'TS29122_CommonData.yaml#/components/responses/413'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'

  /onboardedInvokers/{onboardingId}:
    delete:
      description: Deletes an individual API Invoker.
      parameters:
        - name: onboardingId
          in: path
          description: String identifying an individual on-boarded API invoker resource
          required: true
          schema:
            type: string
      responses:
        '204':
          description: The individual API Invoker matching onboardingId was offboarded.
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'

components:
  schemas:
    OnboardingInformation:
      type: object
      properties:
        apiInvokerPublicKey:
          type: string
          description: The API Invoker’s public key
        apiInvokerCertificate:
          type: string
```

```
          description: The API Invoker's generic client certificate
      required:
        - apiInvokerPublicKey
    OnboardingNotificationDestination:
      type: object
      properties:
        notificationDestination:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Uri'
        requestTestNotification:
          type: boolean
          description: Set to true by Subscriber to request the CAPIF core function to send a test
notification as defined in in subclause 7.6. Set to false or omitted otherwise.
        websockNotifConfig:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/WebsockNotifConfig'
    APIList:
      type: array
      items:
        $ref: 'TS29222_CAPIF_Publish_Service_API.yaml#/components/schemas/ServiceAPIDescription'
      minItems: 1
      description: The list of service APIs that the API Invoker is allowed to invoke
    APIInvokerEnrolmentDetails:
      type: object
      properties:
        apiInvokerId:
          type: string
          description: API invoker ID assigned by the CAPIF core function to the API invoker while
on-boarding the API invoker. Shall not be present in the HTTP POST request from the API invoker to
the CAPIF core function, to on-board itself. Shall be present in all other HTTP requests and
responses.
        onboardingInformation:
          $ref: '#/components/schemas/OnboardingInformation'
        onboardingNotificationDestination:
          $ref: '#/components/schemas/OnboardingNotificationDestination'
        apiList:
          $ref: '#/components/schemas/APIList'
        apiInvokerInformation:
          type: string
          description: Generic information related to the API invoker such as details of the device
or the application.
      required:
        - onboardingInformation
        - onboardingNotificationDestination
      description: Information about the API Invoker that requested to onboard
    OnboardingRequestAck:
      type: object
      properties:
        onboardingNotificationDestination:
          $ref: '#/components/schemas/OnboardingNotificationDestination'
      required:
        - onboardingNotificationDestination
    OnboardingNotification:
      type: object
      properties:
        result:
          type: boolean
          description: Set to "true" indicate successful on-boarding. Otherwise set to "false"
        resourceLocation:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Uri'
        apiInvokerEnrolmentDetails:
          $ref: '#/components/schemas/APIInvokerEnrolmentDetails'
        apiList:
          $ref: '#/components/schemas/APIList'
      required:
        - result
```

A.6 CAPIF_Security_API

```
openapi: 3.0.0
info:
  title: CAPIF_Security_API
  version: "1.PreR15.0.0"
servers:
  - url: '{apiRoot}/capif-security/v1'
    variables:
      apiRoot:
        default: https://demohost.com
        description: apiRoot as defined in subclause 7.5 of 3GPP TS 29.222.
```

```
paths:
  /trustedInvokers:
    post:
      requestBody:
        description: All trusted API invokers
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/ServiceSecurity'
      callbacks:
        notificationDestination:
          '{request.body#/notificationDestination}':
            post:
              requestBody:
                required: true
                content:
                  application/json:
                    schema:
                      $ref: '#/components/schemas/SecurityNotification'
              responses:
                '204':
                  description: No Content (successful notification)
                '400':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/400'
                '401':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/401'
                '403':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/403'
                '409':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/409'
                '411':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/411'
                '412':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/412'
                '413':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/413'
                '500':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/500'
                '503':
                  $ref: 'TS29122_CommonData.yaml#/components/responses/503'
                default:
                  $ref: 'TS29122_CommonData.yaml#/components/responses/default'
      responses:
        '201':
          description: Successful created.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/ServiceSecurity'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '411':
          $ref: 'TS29122_CommonData.yaml#/components/responses/411'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '413':
          $ref: 'TS29122_CommonData.yaml#/components/responses/413'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'

  /trustedInvokers/{apiInvokerId}:
    get:
      parameters:
        - name: apiInvokerId
          in: path
          description: Identifier of an individual API invoker
          required: true
```

```
          schema:
            type: string
        - name: authenticationInfo
          in: query
          description: When set to 'true', it indicates the CAPIF core function to send the
authentication information of the API invoker. Set to false or omitted otherwise.
          schema:
            type: boolean
        - name: authorizationInfo
          in: query
          description: When set to 'true', it indicates the CAPIF core function to send the
authorization information of the API invoker. Set to false or omitted otherwise.
          schema:
            type: boolean
      responses:
        '200':
          description: The security related information of the API Invoker based on the request from
the API exposing function.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/ServiceSecurity'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
    delete:
      parameters:
        - name: apiInvokerId
          in: path
          description: Identifier of an individual API invoker
          required: true
          schema:
            type: string
      responses:
        '204':
          description: No Content (Successful deletion of the existing subscription)
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'

components:
  schemas:
    ServiceSecurity:
      type: object
      properties:
        securityPreferences:
          type: array
          items:
            $ref: '#/components/schemas/SecurityMethod'
          minimum: 0
        selectedSecurityMethods:
          type: array
```

```
          items:
            $ref: '#/components/schemas/SecurityMethod'
          minimum: 0
        apiInvokerSecurityMethods:
          type: array
          items:
            $ref: '#/components/schemas/SecurityMethod'
          minItems: 0
        securityNotificationDestination:
          $ref: '#/components/schemas/SecurityNotificationDestination'
      required:
        - securityNotificationDestination
    SecurityMethod:
      type: object
      properties:
        interfaceDetails:
          $ref: 'TS29222_CAPIF_Publish_Service_API.yaml#/components/schemas/InterfaceDescription'
        securityMethod:
          type: string
          description: Security method for the interface
        authenticationInfo:
          type: string
          description: Authentication related information
        authorizationInfo:
          type: string
          description: Authorization related information
      required:
        - interfaceDetails
        - securityMethod
    SecurityNotificationDestination:
      type: object
      properties:
        notificationDestination:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Uri'
        requestTestNotification:
          type: boolean
          description: Set to true by API invoker to request the CAPIF core function to send a test
notification as defined in in subclause 7.6. Set to false or omitted otherwise.
        websockNotifConfig:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/WebsockNotifConfig'
      required:
        - notificationDestination
    SecurityNotification:
      type: object
      properties:
        apiInvokerId:
          type: string
          description: String identifying the API invoker assigned by the CAPIF core function
        apiId:
          type: string
          description: Identifier of the service API
        cause:
          $ref: '#/components/schemas/Cause'
      required:
        - apiInvokerId
        - apiId
        - cause
    Cause:
      anyOf:
      - type: string
        enum:
          - OVERLIMIT_USAGE
          - UNEXPECTED_REASON
      - type: string
        description: >
          This string provides forward-compatibility with future
          extensions to the enumeration but is not used to encode
          content defined in the present version of this API.
      description: >
        Possible values are
        - OVERLIMIT_USAGE: The revocation of the authorization of the API invoker is due to the
overlimit usage of the service API
        - UNEXPECTED_REASON: The revocation of the authorization of the API invoker is due to
unexpected reason.
```

A.7 CAPIF_Access_Control_Policy_API

```
openapi: 3.0.0
info:
  title: CAPIF_Access_Control_Policy_API
  version: "1.PreR15.0.0"
servers:
  - url: '{apiRoot}/access-control-policy/v1'
    variables:
      apiRoot:
        default: https://demohost.com
        description: apiRoot as defined in subclause 7.5 of 3GPP TS 29.222

paths:
  /accessControlPolicyList/{serviceApiId}:
    get:
      description: Retrieves the access control policy list.
      parameters:
        - name: serviceApiId
          in: path
          description: Identifier of a published service API
          required: true
          schema:
            type: string
        - name: apiInvokerId
          in: query
          description: Identifier of the API invoker
          schema:
            type: string
      responses:
        '200':
          description: OK.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/accessControlPolicyList'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '404':
          $ref: 'TS29122_CommonData.yaml#/components/responses/404'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '414':
          $ref: 'TS29122_CommonData.yaml#/components/responses/414'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'

components:
  schemas:
    accessControlPolicyList:
      type: object
      properties:
        apiInvokerPolicies:
          type: array
          items:
            $ref: '#/components/schemas/ApiInvokerPolicy'
          minItems: 0
          description: Policy of each API invoker.
    ApiInvokerPolicy:
      type: object
      properties:
        apiInvokerId:
          type: string
          description: API invoker ID assigned by the CAPIF core function
        allowedTotalInvocations:
          type: integer
          description: Total number of invocations allowed on the service API by the API invoker.
        allowedInvocationsPerSecond:
          type: integer
```

```
          description: Invocations per second allowed on the service API by the API invoker.
        allowedInvocationTimeRangeList:
          type: array
          items:
            $ref: '#/components/schemas/TimeRangeList'
          minItems: 0
          description: The time ranges during which the invocations are allowed on the service API
by the API invoker.
      required:
        - apiInvokerID
    TimeRangeList:
      type: object
      properties:
        startTime:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/DateTime'
        stopTime:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/DateTime'
```

A.8 CAPIF_Logging_API_Invocation_API

```
openapi: 3.0.0
info:
  title: CAPIF_Logging_API_Invocation_API
  version: "1.PreR15.0.0"
servers:
  - url: '{apiRoot}/api-invocation-logs/v1'
    variables:
      apiRoot:
        default: https://demohost.com
        description: apiRoot as defined in subclause 7.5 of 3GPP TS 29.222
paths:
  /{aefId}/logs:
    post:
      description: Creates a new log entry for service API invocations.
      parameters:
        - name: aefId
          in: path
          description: Identifier of the API exposing function
          required: true
          schema:
            type: string
      requestBody:
        required: true
        content:
          application/json:
            schema:
              $ref: '#/components/schemas/InvocationLogs'
      responses:
        '201':
          description: Log of service API invocations provided by API exposing function successfully
stored on the CAPIF core function.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/InvocationLogs'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '411':
          $ref: 'TS29122_CommonData.yaml#/components/responses/411'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '413':
          $ref: 'TS29122_CommonData.yaml#/components/responses/413'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
  /{aefId}/logs/{logId}:
    description: Creates a new log entry for service API invocations.
```

```
    parameters:
      - name: aefId
        in: path
        description: Identifier of the API exposing function
        required: true
        schema:
          type: string
      - name: logId
        in: path
        description: Identifier of individual log entry
        required: true
        schema:
          type: string
components:
  schemas:
    InvocationLogs:
      type: object
      properties:
        aefId:
          type: string
          description: Identity information of the API exposing function requesting logging of
service API invocations
        apiInvokerId:
          type: string
          description: Identity of the API invoker which invoked the service API
        ipv4Addr:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Ipv4Addr'
        ipv6Addr:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Ipv6Addr'
        port:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/Port'
        logs:
          type: array
          items:
            $ref: '#/components/schemas/Log'
          minItems: 1
          description: Service API invocation log
      required:
        - aefId
        - apiInvokerId
        - log
    Log:
      type: object
      properties:
        apiId:
          type: string
          description: String identifying the API invoked.
        apiInvoked:
          type: string
          description: Name of the API which was invoked
        version:
          type: number
          description: Version of the API which was invoked
        resourceName:
          type: string
          description: Name of the specific resource invoked
        operation:
          type: string
          description: Operation that was invoked on the API
        result:
          type: string
          description: Result or output of the invocation
        invocationTime:
          $ref: 'TS29122_CommonData.yaml#/components/schemas/DateTime'
        parameters:
          type: string
          description: List of input parameters
        interfaceDescription:
          $ref: 'TS29222_CAPIF_Publish_Service_API.yaml/InterfaceDescription'
      required:
        - apiId
        - apiInvoked
        - version
        - resourceName
        - operation
        - result
```

A.9 CAPIF_Auditing_API

```
openapi: 3.0.0
info:
  description: Query_API_Invocation_Log
  title: CAPIF_Auditing_API
  version: "1.PreR15.0.0"
servers:
  - url: '{apiRoot}/service-apis/v1'
    variables:
      apiRoot:
        default: https://demohost.com
        description: apiRoot as defined in subclause 7.5 of 3GPP TS 29.222.
paths:
  /apiInvocationLogs:
    get:
      description: Query and retrieve service API invocation logs stored on the CAPIF core function.
      parameters:
        - name: aefId
          in: query
          description: String identifying the API exposing function.
          schema:
            type: string
        - name: apiInvokerId
          in: query
          description: String identifying the API invoker which invoked the service API.
          schema:
            type: string
        - name: ipv4Addr
          in: query
          description: String identifying a IPv4 address of the API invoker. This attribute shall
not be present if ipv6Addr attribute is present.
          schema:
            $ref: 'TS29122_CommonData.yaml#/components/schemas/Ipv4Addr'
        - name: ipv6Addr
          in: query
          description: String identifying a IPv4 address of the API invoker. This attribute shall
not be present if ipv6Addr attribute is present.
          schema:
            $ref: 'TS29122_CommonData.yaml#/components/schemas/Ipv6Addr'
        - name: port
          in: query
          description: Port. This attribute shall be present if either ipv4Addr or the ipv6Addr
attribute is present.
          schema:
            $ref: 'TS29122_CommonData.yaml#/components/schemas/Port'
        - name: timeRangeStart
          in: query
          description: Start time of the invocation time range.
          schema:
            $ref: 'TS29122_CommonData.yaml#/components/schemas/DateTime'
        - name: timeRangeEnd
          in: query
          description: End time of the invocation time range.
          schema:
            $ref: 'TS29122_CommonData.yaml#/components/schemas/DateTime'
        - name: apiId
          in: query
          description: String identifying the API invoked.
          schema:
            type: string
        - name: apiName
          in: query
          description: Name of the API which was invoked.
          schema:
            type: string
        - name: version
          in: query
          description: Version of the API which was invoked.
          schema:
            type: number
        - name: operation
          in: query
          description: Operation that was invoked on the API.
          schema:
            type: string
        - name: result
```

```
          in: query
          description: Result or output of the invocation.
          schema:
            type: string
        - name: resourceName
          in: query
          description: Name of the specific resource invoked.
          schema:
            type: string
        - name: interfaceDescription
          in: query
          description: Interface description of the API invoked.
          schema:
            $ref: 'TS29222_CAPIF_Publish_Service_API.yaml#/components/schemas/InterfaceDescription'
      responses:
        '200':
          description: Result of the query operation along with fetched service API invocation log
data.
          content:
            application/json:
              schema:
                $ref: '#/components/schemas/InvocationLogs'
        '400':
          $ref: 'TS29122_CommonData.yaml#/components/responses/400'
        '401':
          $ref: 'TS29122_CommonData.yaml#/components/responses/401'
        '403':
          $ref: 'TS29122_CommonData.yaml#/components/responses/403'
        '409':
          $ref: 'TS29122_CommonData.yaml#/components/responses/409'
        '412':
          $ref: 'TS29122_CommonData.yaml#/components/responses/412'
        '414':
          description: URI Too Long
          content:
            application/problem+json:
              schema:
                $ref: 'TS29122_CommonData.yaml#/components/schemas/ProblemDetails'
        '500':
          $ref: 'TS29122_CommonData.yaml#/components/responses/500'
        '503':
          $ref: 'TS29122_CommonData.yaml#/components/responses/503'
        default:
          $ref: 'TS29122_CommonData.yaml#/components/responses/default'
components:
  schemas:
    InvocationLogs:
      type: array
      items:
        $ref: 'TS29222_CAPIF_Logging_API_Invocation.yaml#/components/schemas/InvocationLogs'
```

A.10 AEF_Authentication_API

```
openapi: 3.0.0
info:
  title: AEF_Authentication_API
  version: "1.PreR15.0.0"
servers:
  - url: '{apiRoot}/auth-initiation/v1'
    variables:
      apiRoot:
        default: https://demohost.com
        description: apiRoot as defined in subclause 7.5 of 3GPP TS 29.222.
paths:
  /auth-profiles:
    get:
      parameters:
        - name: apiInvokerId
          in: query
          description: API invoker ID assigned by the CAPIF core function to the API invoker while
on-boarding the API invoker.
          schema:
            type: string
      responses:
        '204':
          description: API exposing function confirms that the authentication profile is available
and it is ready for authentication of the API invoker.
```

```
'400':
  $ref: 'TS29122_CommonData.yaml#/components/responses/400'
'401':
  $ref: 'TS29122_CommonData.yaml#/components/responses/401'
'403':
  $ref: 'TS29122_CommonData.yaml#/components/responses/403'
'404':
  $ref: 'TS29122_CommonData.yaml#/components/responses/404'
'412':
  $ref: 'TS29122_CommonData.yaml#/components/responses/412'
'414':
  description: URI Too Long
  content:
    application/problem+json:
      schema:
        $ref: 'TS29122_CommonData.yaml#/components/schemas/ProblemDetails'
'500':
  $ref: 'TS29122_CommonData.yaml#/components/responses/500'
'503':
  $ref: 'TS29122_CommonData.yaml#/components/responses/503'
default:
  $ref: 'TS29122_CommonData.yaml#/components/responses/default'
```

Annex B (informative): Change history

Change history							
Date	**Meeting**	**TDoc**	**CR**	**Rev**	**Cat**	**Subject/Comment**	**New version**
2018-03	CT3#95	C3-181278				TS skeleton of Common API Framework for 3GPP Northbound APIs	0.0.0
2018-03	CT3#95	C3-181378				Inclusion of documents agreed in CT3#95: C3-181281, C3-181282, C3-181283, C3-181284, C3-181285, C3-181286, C3-181287, C3-181321, C3-181322, Rapporteur changes	0.1.0
2018-04	CT3#96	C3-182527				Inclusion of documents agreed in CT3#96: C3-182204, C3-182387, C3-182393, C3-182395, C3-182468, C3-182469, C3-182470, C3-182483, C3-182484, C3-182485	0.2.0
2018-05	CT3#97					Inclusion of documents agreed in CT3#97: C3-183271, C3-183274, C3-183275, C3-183372, C3-183376, C3-183377, C3-183378, C3-183379, C3-183598, C3-183599, C3-183602, C3-183603, C3-183604, C3-183798, C3-183799, C3-183809, C3-183841, C3-183842	0.3.0
2018-06	CT#80	CP-181037				TS sent to plenary for approval	1.0.0
2018-06	CT#80	CP-181037				TS approved by plenary	15.0.0
2018-09	CT#81	CP-182016	0001	1	F	Changes to clause 4 – Overview	15.1.0
2018-09	CT#81	CP-182016	0003	2	F	Changes to CAPIF Publish Service API subclause	15.1.0
2018-09	CT#81	CP-182016	0004	2	F	Changes to CAPIF Events API subclause	15.1.0
2018-09	CT#81	CP-182016	0005	4	F	Changes to CAPIF API Invoker Management API subclause	15.1.0
2018-09	CT#81	CP-182016	0006	4	F	Changes to CAPIF Authentication Authorization API subclause	15.1.0
2018-09	CT#81	CP-182016	0007	3	F	Update to data types for ServiceAPIDescription and APIQuery	15.1.0
2018-09	CT#81	CP-182016	0008	5	F	Definition of CAPIF_Access_Control_Policy_API, and OpenAPI schema	15.1.0
2018-09	CT#81	CP-182016	0009	4	F	CAPIF_Events_API OpenAPI schema	15.1.0
2018-09	CT#81	CP-182016	0010	4	F	AEF_Authentication_API OpenAPI schema	15.1.0
2018-09	CT#81	CP-182016	0011	1	F	CAPIF_Discover_Service API - Corrections	15.1.0
2018-09	CT#81	CP-182016	0012	3	F	CAPIF_discovery_service API OpenAPI file	15.1.0
2018-09	CT#81	CP-182016	0013	4	F	CAPIF_Publish_Service API - Corrections and OpenAPI file	15.1.0
2018-09	CT#81	CP-182016	0014	4	F	AEF_Authentication API - Editor's notes	15.1.0
2018-09	CT#81	CP-182016	0015	4	F	Corrections to data type	15.1.0
2018-09	CT#81	CP-182016	0016	1	F	API Invoker's Information in APIInvokerEnrolmentDetails	15.1.0
2018-09	CT#81	CP-182016	0017	1	F	Corrections to OnboardingInformation data type	15.1.0
2018-09	CT#81	CP-182016	0018	2	F	Security method preference	15.1.0
2018-09	CT#81	CP-182016	0019	1	F	Clarifications to Obtain_API_Invoker_Info service operation	15.1.0
2018-09	CT#81	CP-182016	0020	1	F	Subscribed and Subscribing functional entity	15.1.0
2018-09	CT#81	CP-182016	0021	1	F	Miscellaneous corrections	15.1.0
2018-09	CT#81	CP-182016	0023	1	F	Definitions and abbreviations	15.1.0
2018-09	CT#81	CP-182016	0024	1	F	Referenced data types and enumerations	15.1.0
2018-09	CT#81	CP-182016	0025	2	F	CAPIF_Security_API OpenAPI schema	15.1.0
2018-09	CT#81	CP-182016	0026	1	F	CAPIF discovery service API – API invoker retrieves API information using GET	15.1.0
2018-09	CT#81	CP-182016	0028	2	F	CAPIF_Auditing_API – API management function retrieves API information logs using GET – OpenAPI document	15.1.0
2018-09	CT#81	CP-182016	0029	3	F	API Names changes in clause 5	15.1.0
2018-09	CT#81	CP-182016	0030	-	F	Change security-related API names in clause 8 and 10	15.1.0

2018-09	CT#81	CP-182016	0031	2	F	Describe response code 202 for Onboard_API_Invoker POST method	15.1.0
2018-09	CT#81	CP-182016	0032	-	F	Correct cardinality for onboardingNotificationDestination	15.1.0
2018-09	CT#81	CP-182016	0033	-	F	Correct cardinality for securityNotificationDestination	15.1.0
2018-09	CT#81	CP-182016	0034	1	F	Correct protocol type in Interface Description	15.1.0
2018-09	CT#81	CP-182016	0036	1	F	Query parameter in retrieving access control	15.1.0
2018-09	CT#81	CP-182037	0037	1	F	Authorization endpoint and token request	15.1.0
2018-09	CT#81	CP-182016	0038	1	F	CAPIF Events	15.1.0
2018-09	CT#81	CP-182016	0040	1	F	Corrections to resource figures	15.1.0
2018-09	CT#81	CP-182016	0041	1	F	CAPIF_Auditing_API - 'query' custom operation	15.1.0
2018-09	CT#81	CP-182016	0042	2	F	OpenAPI - CAPIF_API_Invoker_Management API	15.1.0
2018-09	CT#81	CP-182016	0043	2	F	OpenAPI - CAPIF_Logging_API_Invocation API	15.1.0

History

Document history		
V15.0.0	July 2018	Publication
V15.1.0	October 2018	Publication