3GPP TS 29.222 V16.11.0 (2023-03)

Technical Specification

3rd Generation Partnership Project;

Technical Specification Group Core Network and Terminals;

Common API Framework for 3GPP Northbound APIs;

(Release 16)

** 

The present document has been developed within the 3rd Generation Partnership Project (3GPP TM) and may be further elaborated for the purposes of 3GPP..  
The present document has not been subject to any approval process by the 3GPPOrganizational Partners and shall not be implemented.  
This Specification is provided for future development work within 3GPPonly. The Organizational Partners accept no liability for any use of this Specification.  
Specifications and Reports for implementation of the 3GPP TM system should be obtained via the 3GPP Organizational Partners' Publications Offices.

Keywords

5G System

***3GPP***

Postal address

3GPP support office address

650 Route des Lucioles - Sophia Antipolis

Valbonne - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Internet

http://www.3gpp.org

***Copyright Notification***

No part may be reproduced except as authorized by written permission.  
The copyright and the foregoing restriction extend to reproduction in all media.

© 2023, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).

All rights reserved.

UMTS™ is a Trade Mark of ETSI registered for the benefit of its members

3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners  
LTE™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners

GSM® and the GSM logo are registered and owned by the GSM Association

Contents

Foreword [12](#__RefHeading___Toc120698504)

1 Scope [13](#__RefHeading___Toc120698505)

2 References [13](#__RefHeading___Toc120698506)

3 Definitions and abbreviations [14](#__RefHeading___Toc120698507)

3.1 Definitions [14](#__RefHeading___Toc120698508)

3.2 Abbreviations [14](#__RefHeading___Toc120698509)

4 Overview [15](#__RefHeading___Toc120698510)

4.1 Introduction [15](#__RefHeading___Toc120698511)

4.2 Service Architecture [15](#__RefHeading___Toc120698512)

4.3 Functional Entities [17](#__RefHeading___Toc120698513)

4.3.1 API invoker [17](#__RefHeading___Toc120698514)

4.3.2 CAPIF core function [17](#__RefHeading___Toc120698515)

4.3.3 API exposing function [18](#__RefHeading___Toc120698516)

4.3.4 API publishing function [18](#__RefHeading___Toc120698517)

4.3.5 API management function [18](#__RefHeading___Toc120698518)

5 Services offered by the CAPIF Core Function [19](#__RefHeading___Toc120698519)

5.1 Introduction of Services [19](#__RefHeading___Toc120698520)

5.2 CAPIF\_Discover\_Service\_API [20](#__RefHeading___Toc120698521)

5.2.1 Service Description [20](#__RefHeading___Toc120698522)

5.2.1.1 Overview [20](#__RefHeading___Toc120698523)

5.2.2 Service Operations [20](#__RefHeading___Toc120698524)

5.2.2.1 Introduction [20](#__RefHeading___Toc120698525)

5.2.2.2 Discover\_Service\_API [21](#__RefHeading___Toc120698526)

5.2.2.2.1 General [21](#__RefHeading___Toc120698527)

5.2.2.2.2 Consumer discovering service API using Discover\_Service\_API service operation [21](#__RefHeading___Toc120698528)

5.3 CAPIF\_Publish\_Service\_API [21](#__RefHeading___Toc120698529)

5.3.1 Service Description [21](#__RefHeading___Toc120698530)

5.3.1.1 Overview [21](#__RefHeading___Toc120698531)

5.3.2 Service Operations [21](#__RefHeading___Toc120698532)

5.3.2.1 Introduction [21](#__RefHeading___Toc120698533)

5.3.2.2 Publish\_Service\_API [22](#__RefHeading___Toc120698534)

5.3.2.2.1 General [22](#__RefHeading___Toc120698535)

5.3.2.2.2 API publishing function publishing service APIs on CAPIF core function using Publish\_Service\_API service operation [22](#__RefHeading___Toc120698536)

5.3.2.2.3 CAPIF core function publishing service APIs on other CAPIF core function using Publish\_Service\_API service operation [23](#__RefHeading___Toc120698537)

5.3.2.3 Unpublish\_Service\_API [23](#__RefHeading___Toc120698538)

5.3.2.3.1 General [23](#__RefHeading___Toc120698539)

5.3.2.3.2 Consumer un-publishing service APIs from CAPIF core function using Unpublish\_Service\_API service operation [24](#__RefHeading___Toc120698540)

5.3.2.4 Get\_Service\_API [24](#__RefHeading___Toc120698541)

5.3.2.4.1 General [24](#__RefHeading___Toc120698542)

5.3.2.4.2 Consumer retrieving service APIs from CAPIF core function using Get\_Service\_API service operation [24](#__RefHeading___Toc120698543)

5.3.2.5 Update\_Service\_API [25](#__RefHeading___Toc120698544)

5.3.2.5.1 General [25](#__RefHeading___Toc120698545)

5.3.2.5.2 Consumer updating published service APIs on CAPIF core function using Update\_Service\_API service operation [25](#__RefHeading___Toc120698546)

5.4 CAPIF\_Events\_API [25](#__RefHeading___Toc120698547)

5.4.1 Service Description [25](#__RefHeading___Toc120698548)

5.4.1.1 Overview [25](#__RefHeading___Toc120698549)

5.4.2 Service Operations [26](#__RefHeading___Toc120698550)

5.4.2.1 Introduction [26](#__RefHeading___Toc120698551)

5.4.2.2 Subscribe\_Event [26](#__RefHeading___Toc120698552)

5.4.2.2.1 General [26](#__RefHeading___Toc120698553)

5.4.2.2.2 Subscribing to CAPIF events using Subscribe\_Event service operation [26](#__RefHeading___Toc120698554)

5.4.2.3 Unsubscribe\_Event [27](#__RefHeading___Toc120698555)

5.4.2.3.1 General [27](#__RefHeading___Toc120698556)

5.4.2.3.2 Unsubscribing from CAPIF events using Unsubscribe\_Event service operation [27](#__RefHeading___Toc120698557)

5.4.2.4 Notify\_Event [27](#__RefHeading___Toc120698558)

5.4.2.4.1 General [27](#__RefHeading___Toc120698559)

5.4.2.4.2 Notifying CAPIF events using Notify\_Event service operation [27](#__RefHeading___Toc120698560)

5.5 CAPIF\_API\_Invoker\_Management\_API [28](#__RefHeading___Toc120698561)

5.5.1 Service Description [28](#__RefHeading___Toc120698562)

5.5.1.1 Overview [28](#__RefHeading___Toc120698563)

5.5.2 Service Operations [28](#__RefHeading___Toc120698564)

5.5.2.1 Introduction [28](#__RefHeading___Toc120698565)

5.5.2.2 Onboard\_API\_Invoker [28](#__RefHeading___Toc120698566)

5.5.2.2.1 General [28](#__RefHeading___Toc120698567)

5.5.2.2.2 API invoker on-boarding itself as a recognized user of CAPIF using Onboard\_API\_Invoker service operation [28](#__RefHeading___Toc120698568)

5.5.2.3 Offboard\_API\_Invoker [29](#__RefHeading___Toc120698569)

5.5.2.3.1 General [29](#__RefHeading___Toc120698570)

5.5.2.3.2 API invoker off-boarding itself as a recognized user of CAPIF using Offboard\_API\_Invoker service operation [29](#__RefHeading___Toc120698571)

5.5.2.4 Notify\_Onboarding\_Completion [30](#__RefHeading___Toc120698572)

5.5.2.4.1 General [30](#__RefHeading___Toc120698573)

5.5.2.4.2 Notifying Onboarding completion using Notify\_Onboarding\_Completion service operation [30](#__RefHeading___Toc120698574)

5.5.2.5 Update\_API\_Invoker\_Details [30](#__RefHeading___Toc120698575)

5.5.2.5.1 General [30](#__RefHeading___Toc120698576)

5.5.2.5.2 API invoker updating its details on CAPIF using Update\_API\_Invoker\_Details service operation [30](#__RefHeading___Toc120698577)

5.5.2.6 Notify\_Update\_Completion [31](#__RefHeading___Toc120698578)

5.5.2.6.1 General [31](#__RefHeading___Toc120698579)

5.5.2.6.2 Notifying API invoker update completion using Notify\_Update\_Completion service operation [31](#__RefHeading___Toc120698580)

5.6 CAPIF\_Security\_API [31](#__RefHeading___Toc120698581)

5.6.1 Service Description [31](#__RefHeading___Toc120698582)

5.6.1.1 Overview [31](#__RefHeading___Toc120698583)

5.6.2 Service Operations [32](#__RefHeading___Toc120698584)

5.6.2.1 Introduction [32](#__RefHeading___Toc120698585)

5.6.2.2 Obtain\_Security\_Method [32](#__RefHeading___Toc120698586)

5.6.2.2.1 General [32](#__RefHeading___Toc120698587)

5.6.2.2.2 Request service API security method from CAPIF using Obtain\_Security\_Method service operation [32](#__RefHeading___Toc120698588)

5.6.2.3 Obtain\_Authorization [33](#__RefHeading___Toc120698589)

5.6.2.3.1 General [33](#__RefHeading___Toc120698590)

5.6.2.3.2 Obtain authorization using Obtain\_Authorization service operation [33](#__RefHeading___Toc120698591)

5.6.2.4 Obtain\_API\_Invoker\_Info [33](#__RefHeading___Toc120698592)

5.6.2.4.1 General [33](#__RefHeading___Toc120698593)

5.6.2.4.2 Obtain API invoker's security information using Obtain\_API\_Invoker\_Info service operation [33](#__RefHeading___Toc120698594)

5.6.2.5 Revoke\_Authentication [34](#__RefHeading___Toc120698595)

5.6.2.5.1 General [34](#__RefHeading___Toc120698596)

5.6.2.5.2 Invalidate authorization using Revoke\_Authorization service operation [34](#__RefHeading___Toc120698597)

5.7 CAPIF\_Monitoring\_API [34](#__RefHeading___Toc120698598)

5.8 CAPIF\_Logging\_API\_Invocation\_API [34](#__RefHeading___Toc120698599)

5.8.1 Service Description [34](#__RefHeading___Toc120698600)

5.8.1.1 Overview [34](#__RefHeading___Toc120698601)

5.8.2 Service Operations [35](#__RefHeading___Toc120698602)

5.8.2.1 Introduction [35](#__RefHeading___Toc120698603)

5.8.2.2 Log\_API\_Invocation\_API [35](#__RefHeading___Toc120698604)

5.8.2.2.1 General [35](#__RefHeading___Toc120698605)

5.8.2.2.2 Logging service API invocations using Log\_API\_Invocation service operation [35](#__RefHeading___Toc120698606)

5.9 CAPIF\_Auditing\_API [35](#__RefHeading___Toc120698607)

5.9.1 Service Description [35](#__RefHeading___Toc120698608)

5.9.1.1 Overview [35](#__RefHeading___Toc120698609)

5.9.2 Service Operations [36](#__RefHeading___Toc120698610)

5.9.2.1 Introduction [36](#__RefHeading___Toc120698611)

5.9.2.2 Query\_Invocation\_Logs\_API [36](#__RefHeading___Toc120698612)

5.9.2.2.1 General [36](#__RefHeading___Toc120698613)

5.9.2.2.2 Query API invocation information logs using Query\_Invocation\_Logs service operation [36](#__RefHeading___Toc120698614)

5.10 CAPIF\_Access\_Control\_Policy\_API [36](#__RefHeading___Toc120698615)

5.10.1 Service Description [36](#__RefHeading___Toc120698616)

5.10.1.1 Overview [36](#__RefHeading___Toc120698617)

5.10.2 Service Operations [37](#__RefHeading___Toc120698618)

5.10.2.1 Introduction [37](#__RefHeading___Toc120698619)

5.10.2.2 Obtain\_Access\_Control\_Policy [37](#__RefHeading___Toc120698620)

5.10.2.2.1 General [37](#__RefHeading___Toc120698621)

5.10.2.2.2 API exposing function obtaining access control policy from the CAPIF core function using Obtain\_Access\_Control\_Policy service operation [37](#__RefHeading___Toc120698622)

5.10.3 Related Events [37](#__RefHeading___Toc120698623)

5.11 CAPIF\_API\_Provider\_Management\_API [37](#__RefHeading___Toc120698624)

5.11.1 Service Description [37](#__RefHeading___Toc120698625)

5.11.1.1 Overview [37](#__RefHeading___Toc120698626)

5.11.2 Service Operations [37](#__RefHeading___Toc120698627)

5.11.2.1 Introduction [37](#__RefHeading___Toc120698628)

5.11.2.2 Register\_API\_Provider [38](#__RefHeading___Toc120698629)

5.11.2.2.1 General [38](#__RefHeading___Toc120698630)

5.11.2.2.2 API provider domain functions registering as a recognized API provider domain function of CAPIF using Register\_API\_Provider service operation [38](#__RefHeading___Toc120698631)

5.11.2.3 Update\_API\_Provider [38](#__RefHeading___Toc120698632)

5.11.2.3.1 General [38](#__RefHeading___Toc120698633)

5.11.2.3.2 API management function updating API provider domain function details on CAPIF using Update\_API\_Provider service operation [38](#__RefHeading___Toc120698634)

5.11.2.4 Deregister\_API\_Provider [39](#__RefHeading___Toc120698635)

5.11.2.4.1 General [39](#__RefHeading___Toc120698636)

5.11.2.4.2 API provider domain functions deregistering as a recognized API provider domain function of CAPIF using Deregister\_API\_Provider service operation [39](#__RefHeading___Toc120698637)

5.12 CAPIF\_Routing\_Info\_API [39](#__RefHeading___Toc120698638)

5.12.1 Service Description [39](#__RefHeading___Toc120698639)

5.12.1.1 Overview [39](#__RefHeading___Toc120698640)

5.12.2 Service Operations [40](#__RefHeading___Toc120698641)

5.12.2.1 Introduction [40](#__RefHeading___Toc120698642)

5.12.2.2 Obtain\_Routing\_Info [40](#__RefHeading___Toc120698643)

5.12.2.2.1 General [40](#__RefHeading___Toc120698644)

5.12.2.2.2 API exposing function obtaining API routing information from the CAPIF core function using Obtain\_Routing\_Info service operation [40](#__RefHeading___Toc120698645)

6 Services offered by the API exposing function [40](#__RefHeading___Toc120698646)

6.1 Introduction of Services [40](#__RefHeading___Toc120698647)

6.2 AEF\_Security\_API [41](#__RefHeading___Toc120698648)

6.2.1 Service Description [41](#__RefHeading___Toc120698649)

6.2.1.1 Overview [41](#__RefHeading___Toc120698650)

6.2.2 Service Operations [41](#__RefHeading___Toc120698651)

6.2.2.1 Introduction [41](#__RefHeading___Toc120698652)

6.2.2.2 Initiate\_Authentication [41](#__RefHeading___Toc120698653)

6.2.2.2.1 General [41](#__RefHeading___Toc120698654)

6.2.2.2.2 API invoker initiating authentication using Initiate\_Authentication service operation [41](#__RefHeading___Toc120698655)

6.2.2.3 Revoke\_Authorization [42](#__RefHeading___Toc120698656)

6.2.2.3.1 General [42](#__RefHeading___Toc120698657)

6.2.2.3.2 CAPIF core function initiating revocation using Revoke\_Authorization service operation [42](#__RefHeading___Toc120698658)

7 CAPIF Design Aspects Common for All APIs [42](#__RefHeading___Toc120698659)

7.1 General [42](#__RefHeading___Toc120698660)

7.2 Data Types [42](#__RefHeading___Toc120698661)

7.2.1 General [42](#__RefHeading___Toc120698662)

7.2.2 Referenced structured data types [43](#__RefHeading___Toc120698663)

7.2.3 Referenced Simple data types and enumerations [43](#__RefHeading___Toc120698664)

7.3 Usage of HTTP [43](#__RefHeading___Toc120698665)

7.4 Content type [43](#__RefHeading___Toc120698666)

7.5 URI structure [44](#__RefHeading___Toc120698667)

7.6 Notifications [44](#__RefHeading___Toc120698668)

7.7 Error handling [44](#__RefHeading___Toc120698669)

7.8 Feature negotiation [45](#__RefHeading___Toc120698670)

7.9 HTTP headers [45](#__RefHeading___Toc120698671)

7.10 Conventions for Open API specification files [45](#__RefHeading___Toc120698672)

8 CAPIF API Definition [45](#__RefHeading___Toc120698673)

8.1 CAPIF\_Discover\_Service\_API [45](#__RefHeading___Toc120698674)

8.1.1 API URI [45](#__RefHeading___Toc120698675)

8.1.2 Resources [46](#__RefHeading___Toc120698676)

8.1.2.1 Overview [46](#__RefHeading___Toc120698677)

8.1.2.2 Resource: All published service APIs [46](#__RefHeading___Toc120698678)

8.1.2.2.1 Description [46](#__RefHeading___Toc120698679)

8.1.2.2.2 Resource Definition [46](#__RefHeading___Toc120698680)

8.1.2.2.3 Resource Standard Methods [46](#__RefHeading___Toc120698681)

8.1.2.2.3.1 GET [46](#__RefHeading___Toc120698682)

8.1.2.2.4 Resource Custom Operations [47](#__RefHeading___Toc120698683)

8.1.3 Notifications [47](#__RefHeading___Toc120698684)

8.1.4 Data Model [47](#__RefHeading___Toc120698685)

8.1.4.1 General [47](#__RefHeading___Toc120698686)

8.1.4.2 Structured data types [48](#__RefHeading___Toc120698687)

8.1.4.2.1 Introduction [48](#__RefHeading___Toc120698688)

8.1.4.2.2 Type: DiscoveredAPIs [48](#__RefHeading___Toc120698689)

8.1.4.2.3 Void [48](#__RefHeading___Toc120698690)

8.1.4.3 Simple data types and enumerations [48](#__RefHeading___Toc120698691)

8.1.5 Error Handling [48](#__RefHeading___Toc120698692)

8.1.6 Feature negotiation [48](#__RefHeading___Toc120698693)

8.2 CAPIF\_Publish\_Service\_API [49](#__RefHeading___Toc120698694)

8.2.1 API URI [49](#__RefHeading___Toc120698695)

8.2.2 Resources [49](#__RefHeading___Toc120698696)

8.2.2.1 Overview [49](#__RefHeading___Toc120698697)

8.2.2.2 Resource: APF published APIs [50](#__RefHeading___Toc120698698)

8.2.2.2.1 Description [50](#__RefHeading___Toc120698699)

8.2.2.2.2 Resource Definition [50](#__RefHeading___Toc120698700)

8.2.2.2.3 Resource Standard Methods [50](#__RefHeading___Toc120698701)

8.2.2.2.3.1 POST [50](#__RefHeading___Toc120698702)

8.2.2.2.3.2 GET [51](#__RefHeading___Toc120698703)

8.2.2.2.4 Resource Custom Operations [51](#__RefHeading___Toc120698704)

8.2.2.3 Resource: Individual APF published API [51](#__RefHeading___Toc120698705)

8.2.2.3.1 Description [51](#__RefHeading___Toc120698706)

8.2.2.3.2 Resource Definition [51](#__RefHeading___Toc120698707)

8.2.2.3.3 Resource Standard Methods [51](#__RefHeading___Toc120698708)

8.2.2.3.3.1 GET [51](#__RefHeading___Toc120698709)

8.2.2.3.3.2 PUT [52](#__RefHeading___Toc120698710)

8.2.2.3.3.3 DELETE [52](#__RefHeading___Toc120698711)

8.2.2.3.4 Resource Custom Operations [53](#__RefHeading___Toc120698712)

8.2.3 Notifications [53](#__RefHeading___Toc120698713)

8.2.4 Data Model [53](#__RefHeading___Toc120698714)

8.2.4.1 General [53](#__RefHeading___Toc120698715)

8.2.4.2 Structured data types [54](#__RefHeading___Toc120698716)

8.2.4.2.1 Introduction [54](#__RefHeading___Toc120698717)

8.2.4.2.2 Type: ServiceAPIDescription [54](#__RefHeading___Toc120698718)

8.2.4.2.3 Type: InterfaceDescription [55](#__RefHeading___Toc120698719)

8.2.4.2.4 Type: AefProfile [55](#__RefHeading___Toc120698720)

8.2.4.2.5 Type: Version [55](#__RefHeading___Toc120698721)

8.2.4.2.6 Type: Resource [56](#__RefHeading___Toc120698722)

8.2.4.2.7 Type: CustomOperation [56](#__RefHeading___Toc120698723)

8.2.4.2.8 Type: ShareableInformation [56](#__RefHeading___Toc120698724)

8.2.4.2.9 Type: PublishedApiPath [56](#__RefHeading___Toc120698725)

8.2.4.3 Simple data types and enumerations [57](#__RefHeading___Toc120698726)

8.2.4.3.1 Introduction [57](#__RefHeading___Toc120698727)

8.2.4.3.2 Simple data types [57](#__RefHeading___Toc120698728)

8.2.4.3.3 Enumeration: Protocol [57](#__RefHeading___Toc120698729)

8.2.4.3.4 Enumeration: DataFormat [57](#__RefHeading___Toc120698730)

8.2.4.3.5 Enumeration: CommunicationType [57](#__RefHeading___Toc120698731)

8.2.4.3.6 Enumeration: SecurityMethod [57](#__RefHeading___Toc120698732)

8.2.4.3.7 Enumeration: Operation [58](#__RefHeading___Toc120698733)

8.2.5 Error Handling [58](#__RefHeading___Toc120698734)

8.2.6 Feature negotiation [58](#__RefHeading___Toc120698735)

8.3 CAPIF\_Events\_API [58](#__RefHeading___Toc120698736)

8.3.1 API URI [58](#__RefHeading___Toc120698737)

8.3.2 Resources [59](#__RefHeading___Toc120698738)

8.3.2.1 Overview [59](#__RefHeading___Toc120698739)

8.3.2.2 Resource: CAPIF Events Subscriptions [59](#__RefHeading___Toc120698740)

8.3.2.2.1 Description [59](#__RefHeading___Toc120698741)

8.3.2.2.2 Resource Definition [59](#__RefHeading___Toc120698742)

8.3.2.2.3 Resource Standard Methods [60](#__RefHeading___Toc120698743)

8.3.2.2.3.1 POST [60](#__RefHeading___Toc120698744)

8.3.2.2.4 Resource Custom Operations [60](#__RefHeading___Toc120698745)

8.3.2.3 Resource: Individual CAPIF Events Subscription [60](#__RefHeading___Toc120698746)

8.3.2.3.1 Description [60](#__RefHeading___Toc120698747)

8.3.2.3.2 Resource Definition [60](#__RefHeading___Toc120698748)

8.3.2.3.3 Resource Standard Methods [61](#__RefHeading___Toc120698749)

8.3.2.3.3.1 DELETE [61](#__RefHeading___Toc120698750)

8.3.2.3.4 Resource Custom Operations [61](#__RefHeading___Toc120698751)

8.3.3 Notifications [61](#__RefHeading___Toc120698752)

8.3.3.1 General [61](#__RefHeading___Toc120698753)

8.3.3.2 Event Notification [62](#__RefHeading___Toc120698754)

8.3.3.2.1 Description [62](#__RefHeading___Toc120698755)

8.3.3.2.2 Notification definition [62](#__RefHeading___Toc120698756)

8.3.4 Data Model [62](#__RefHeading___Toc120698757)

8.3.4.1 General [62](#__RefHeading___Toc120698758)

8.3.4.2 Structured data types [63](#__RefHeading___Toc120698759)

8.3.4.2.1 Introduction [63](#__RefHeading___Toc120698760)

8.3.4.2.2 Type: EventSubscription [64](#__RefHeading___Toc120698761)

8.3.4.2.3 Type: EventNotification [64](#__RefHeading___Toc120698762)

8.3.4.2.4 Type: CAPIFEventFilter [64](#__RefHeading___Toc120698763)

8.3.4.2.5 Type: CAPIFEventDetail [65](#__RefHeading___Toc120698764)

8.3.4.2.6 Type: AccessControlPolicyListExt [65](#__RefHeading___Toc120698765)

8.3.4.2.7 Type: TopologyHiding [65](#__RefHeading___Toc120698766)

8.3.4.3 Simple data types and enumerations [65](#__RefHeading___Toc120698767)

8.3.4.3.1 Introduction [65](#__RefHeading___Toc120698768)

8.3.4.3.2 Simple data types [65](#__RefHeading___Toc120698769)

8.3.4.3.3 Enumeration: CAPIFEvent [66](#__RefHeading___Toc120698770)

8.3.5 Error Handling [66](#__RefHeading___Toc120698771)

8.3.6 Feature negotiation [66](#__RefHeading___Toc120698772)

8.4 CAPIF\_API\_Invoker\_Management\_API [67](#__RefHeading___Toc120698773)

8.4.1 API URI [67](#__RefHeading___Toc120698774)

8.4.2 Resources [67](#__RefHeading___Toc120698775)

8.4.2.1 Overview [67](#__RefHeading___Toc120698776)

8.4.2.2 Resource: On-boarded API invokers [68](#__RefHeading___Toc120698777)

8.4.2.2.1 Description [68](#__RefHeading___Toc120698778)

8.4.2.2.2 Resource Definition [68](#__RefHeading___Toc120698779)

8.4.2.2.3 Resource Standard Methods [68](#__RefHeading___Toc120698780)

8.4.2.2.3.1 POST [68](#__RefHeading___Toc120698781)

8.4.2.3.3.2 PUT [69](#__RefHeading___Toc120698782)

8.4.2.2.4 Resource Custom Operations [69](#__RefHeading___Toc120698783)

8.4.2.3 Resource: Individual On-boarded API Invoker [69](#__RefHeading___Toc120698784)

8.4.2.3.1 Description [69](#__RefHeading___Toc120698785)

8.4.2.3.2 Resource Definition [69](#__RefHeading___Toc120698786)

8.4.2.3.3 Resource Standard Methods [70](#__RefHeading___Toc120698787)

8.4.2.3.3.1 DELETE [70](#__RefHeading___Toc120698788)

8.3.2.3.4 Resource Custom Operations [70](#__RefHeading___Toc120698789)

8.4.3 Notifications [70](#__RefHeading___Toc120698790)

8.4.3.1 General [70](#__RefHeading___Toc120698791)

8.4.3.2 Notify\_Onboarding\_Completion [71](#__RefHeading___Toc120698792)

8.4.3.2.1 Description [71](#__RefHeading___Toc120698793)

8.4.3.2.2 Notification definition [71](#__RefHeading___Toc120698794)

8.4.3.3 Notify\_Update\_Completion [71](#__RefHeading___Toc120698795)

8.4.3.3.1 Description [71](#__RefHeading___Toc120698796)

8.4.3.3.2 Notification definition [71](#__RefHeading___Toc120698797)

8.4.4 Data Model [72](#__RefHeading___Toc120698798)

8.4.4.1 General [72](#__RefHeading___Toc120698799)

8.4.4.2 Structured data types [73](#__RefHeading___Toc120698800)

8.4.4.2.1 Introduction [73](#__RefHeading___Toc120698801)

8.4.4.2.2 Type: APIInvokerEnrolmentDetails [73](#__RefHeading___Toc120698802)

8.4.4.2.3 Type: Void [73](#__RefHeading___Toc120698803)

8.4.4.2.4 Type: APIList [73](#__RefHeading___Toc120698804)

8.4.4.2.5 Type: OnboardingInformation [74](#__RefHeading___Toc120698805)

8.4.4.2.6 Type: Void [74](#__RefHeading___Toc120698806)

8.4.4.2.7 Type: OnboardingNotification [74](#__RefHeading___Toc120698807)

8.4.4.3 Simple data types and enumerations [74](#__RefHeading___Toc120698808)

8.4.5 Error Handling [74](#__RefHeading___Toc120698809)

8.4.6 Feature negotiation [74](#__RefHeading___Toc120698810)

8.5 CAPIF\_Security\_API [75](#__RefHeading___Toc120698811)

8.5.1 API URI [75](#__RefHeading___Toc120698812)

8.5.2 Resources [75](#__RefHeading___Toc120698813)

8.5.2.1 Overview [75](#__RefHeading___Toc120698814)

8.5.2.2 Resource: Trusted API invokers [76](#__RefHeading___Toc120698815)

8.5.2.2.1 Description [76](#__RefHeading___Toc120698816)

8.5.2.2.2 Resource Definition [76](#__RefHeading___Toc120698817)

8.5.2.2.3 Resource Standard Methods [76](#__RefHeading___Toc120698818)

8.5.2.2.3.1 Void [76](#__RefHeading___Toc120698819)

8.5.2.2.4 Resource Custom Operations [76](#__RefHeading___Toc120698820)

8.5.2.3 Resource: Individual trusted API invokers [76](#__RefHeading___Toc120698821)

8.5.2.3.1 Description [76](#__RefHeading___Toc120698822)

8.5.2.3.2 Resource Definition [77](#__RefHeading___Toc120698823)

8.5.2.3.3 Resource Standard Methods [77](#__RefHeading___Toc120698824)

8.5.2.3.3.1 GET [77](#__RefHeading___Toc120698825)

8.5.2.3.3.2 DELETE [77](#__RefHeading___Toc120698826)

8.5.2.3.3.3 PUT [78](#__RefHeading___Toc120698827)

8.5.2.3.4 Resource Custom Operations [79](#__RefHeading___Toc120698828)

8.5.2.3.4.1 Overview [79](#__RefHeading___Toc120698829)

8.5.2.3.4.2 Operation: update [79](#__RefHeading___Toc120698830)

8.5.2.3.4.2.1 Description [79](#__RefHeading___Toc120698831)

8.5.2.3.4.2.2 Operation Definition [79](#__RefHeading___Toc120698832)

8.5.2.3.4.3 Operation: delete [80](#__RefHeading___Toc120698833)

8.5.2.3.4.3.1 Description [80](#__RefHeading___Toc120698834)

8.5.2.3.4.3.2 Operation Definition [80](#__RefHeading___Toc120698835)

8.5.2.3.4.4 Operation: token [80](#__RefHeading___Toc120698836)

8.5.2.3.4.4.1 Description [80](#__RefHeading___Toc120698837)

8.5.2.3.4.4.2 Operation Definition [80](#__RefHeading___Toc120698838)

8.5.3 Notifications [81](#__RefHeading___Toc120698839)

8.5.3.1 General [81](#__RefHeading___Toc120698840)

8.5.3.2 Authorization revoked notification [81](#__RefHeading___Toc120698841)

8.5.3.2.1 Description [81](#__RefHeading___Toc120698842)

8.5.3.2.2 Notification definition [81](#__RefHeading___Toc120698843)

8.5.4 Data Model [82](#__RefHeading___Toc120698844)

8.5.4.1 General [82](#__RefHeading___Toc120698845)

8.5.4.2 Structured data types [83](#__RefHeading___Toc120698846)

8.5.4.2.1 Introduction [83](#__RefHeading___Toc120698847)

8.5.4.2.2 Type: ServiceSecurity [83](#__RefHeading___Toc120698848)

8.5.4.2.3 Type: SecurityInformation [84](#__RefHeading___Toc120698849)

8.5.4.2.4 Void [84](#__RefHeading___Toc120698850)

8.5.4.2.5 Type: SecurityNotification [84](#__RefHeading___Toc120698851)

8.5.4.2.6 Type: AccessTokenReq [85](#__RefHeading___Toc120698852)

8.5.4.2.7 Type: AccessTokenRsp [86](#__RefHeading___Toc120698853)

8.5.4.2.8 Type: AccessTokenClaims [86](#__RefHeading___Toc120698854)

8.5.4.3 Simple data types and enumerations [87](#__RefHeading___Toc120698855)

8.5.4.3.1 Introduction [87](#__RefHeading___Toc120698856)

8.5.4.3.2 Simple data types [87](#__RefHeading___Toc120698857)

8.5.4.3.3 Enumeration: Cause [87](#__RefHeading___Toc120698858)

8.5.5 Error Handling [87](#__RefHeading___Toc120698859)

8.5.6 Feature negotiation [87](#__RefHeading___Toc120698860)

8.6 CAPIF\_Access\_Control\_Policy\_API [87](#__RefHeading___Toc120698861)

8.6.1 API URI [87](#__RefHeading___Toc120698862)

8.6.2 Resources [88](#__RefHeading___Toc120698863)

8.6.2.1 Overview [88](#__RefHeading___Toc120698864)

8.6.2.2 Resource: Access Control Policy List [88](#__RefHeading___Toc120698865)

8.6.2.2.1 Description [88](#__RefHeading___Toc120698866)

8.6.2.2.2 Resource Definition [88](#__RefHeading___Toc120698867)

8.6.2.2.3 Resource Standard Methods [89](#__RefHeading___Toc120698868)

8.6.2.2.3.1 GET [89](#__RefHeading___Toc120698869)

8.6.2.2.4 Resource Custom Operations [89](#__RefHeading___Toc120698870)

8.6.3 Notifications [89](#__RefHeading___Toc120698871)

8.6.4 Data Model [89](#__RefHeading___Toc120698872)

8.6.4.1 General [89](#__RefHeading___Toc120698873)

8.6.4.2 Structured data types [90](#__RefHeading___Toc120698874)

8.6.4.2.1 Introduction [90](#__RefHeading___Toc120698875)

8.6.4.2.2 Type: AccessControlPolicyList [90](#__RefHeading___Toc120698876)

8.6.4.2.3 Type: ApiInvokerPolicy [90](#__RefHeading___Toc120698877)

8.6.4.2.4 Type: TimeRangeList [90](#__RefHeading___Toc120698878)

8.6.4.3 Simple data types and enumerations [91](#__RefHeading___Toc120698879)

8.6.5 Error Handling [91](#__RefHeading___Toc120698880)

8.6.6 Feature negotiation [91](#__RefHeading___Toc120698881)

8.7 CAPIF\_Logging\_API\_Invocation\_API [91](#__RefHeading___Toc120698882)

8.7.1 API URI [91](#__RefHeading___Toc120698883)

8.7.2 Resources [92](#__RefHeading___Toc120698884)

8.7.2.1 Overview [92](#__RefHeading___Toc120698885)

8.7.2.2 Resource: Logs [92](#__RefHeading___Toc120698886)

8.7.2.2.1 Description [92](#__RefHeading___Toc120698887)

8.7.2.2.2 Resource Definition [92](#__RefHeading___Toc120698888)

8.7.2.2.3 Resource Standard Methods [93](#__RefHeading___Toc120698889)

8.7.2.2.3.1 POST [93](#__RefHeading___Toc120698890)

8.7.2.2.4 Resource Custom Operations [93](#__RefHeading___Toc120698891)

8.7.3 Notifications [93](#__RefHeading___Toc120698892)

8.7.4 Data Model [93](#__RefHeading___Toc120698893)

8.7.4.1 General [93](#__RefHeading___Toc120698894)

8.7.4.2 Structured data types [94](#__RefHeading___Toc120698895)

8.7.4.2.1 Introduction [94](#__RefHeading___Toc120698896)

8.7.4.2.2 Type: InvocationLog [94](#__RefHeading___Toc120698897)

8.7.4.2.3 Type: Log [95](#__RefHeading___Toc120698898)

8.7.4.3 Simple data types and enumerations [95](#__RefHeading___Toc120698899)

8.7.4.3.1 Introduction [95](#__RefHeading___Toc120698900)

8.7.4.3.2 Simple data types [95](#__RefHeading___Toc120698901)

8.7.5 Error Handling [96](#__RefHeading___Toc120698902)

8.7.6 Feature negotiation [96](#__RefHeading___Toc120698903)

8.8 CAPIF\_Auditing\_API [96](#__RefHeading___Toc120698904)

8.8.1 API URI [96](#__RefHeading___Toc120698905)

8.8.2 Resources [96](#__RefHeading___Toc120698906)

8.8.2.1 Overview [96](#__RefHeading___Toc120698907)

8.8.2.2 Resource: All service API invocation logs [97](#__RefHeading___Toc120698908)

8.8.2.2.1 Description [97](#__RefHeading___Toc120698909)

8.8.2.2.2 Resource Definition [97](#__RefHeading___Toc120698910)

8.8.2.2.3 Resource Standard Methods [97](#__RefHeading___Toc120698911)

8.8.2.2.3.1 GET [97](#__RefHeading___Toc120698912)

8.8.2.2.4 Resource Custom Operations [98](#__RefHeading___Toc120698913)

8.8.3 Notifications [98](#__RefHeading___Toc120698914)

8.8.4 Data Model [98](#__RefHeading___Toc120698915)

8.8.4.1 General [98](#__RefHeading___Toc120698916)

8.8.4.2 Structured data types [98](#__RefHeading___Toc120698917)

8.8.4.3 Simple data types and enumerations [98](#__RefHeading___Toc120698918)

8.8.5 Error Handling [99](#__RefHeading___Toc120698919)

8.8.6 Feature negotiation [99](#__RefHeading___Toc120698920)

8.9 CAPIF\_API\_Provider\_Management\_API [99](#__RefHeading___Toc120698921)

8.9.1 API URI [99](#__RefHeading___Toc120698922)

8.9.2 Resources [99](#__RefHeading___Toc120698923)

8.9.2.1 Overview [99](#__RefHeading___Toc120698924)

8.9.2.2 Resource: All API Provider Domains Registrations [100](#__RefHeading___Toc120698925)

8.9.2.2.1 Description [100](#__RefHeading___Toc120698926)

8.9.2.2.2 Resource Definition [100](#__RefHeading___Toc120698927)

8.9.2.2.3 Resource Standard Methods [100](#__RefHeading___Toc120698928)

8.9.2.2.3.1 POST [100](#__RefHeading___Toc120698929)

8.9.2.2.4 Resource Custom Operations [101](#__RefHeading___Toc120698930)

8.9.2.3 Resource: Individual API Provider Domain Registration [101](#__RefHeading___Toc120698931)

8.9.2.3.1 Description [101](#__RefHeading___Toc120698932)

8.9.2.3.2 Resource Definition [101](#__RefHeading___Toc120698933)

8.9.2.3.3 Resource Standard Methods [101](#__RefHeading___Toc120698934)

8.9.2.3.3.1 PUT [101](#__RefHeading___Toc120698935)

8.9.2.3.3.2 DELETE [102](#__RefHeading___Toc120698936)

8.9.2.3.4 Resource Custom Operations [102](#__RefHeading___Toc120698937)

8.9.3 Notifications [102](#__RefHeading___Toc120698938)

8.9.4 Data Model [103](#__RefHeading___Toc120698939)

8.9.4.1 General [103](#__RefHeading___Toc120698940)

8.9.4.2 Structured data types [104](#__RefHeading___Toc120698941)

8.9.4.2.1 Introduction [104](#__RefHeading___Toc120698942)

8.9.4.2.2 Type: APIProviderEnrolmentDetails [104](#__RefHeading___Toc120698943)

8.9.4.2.3 Type: APIProviderFunctionDetails [105](#__RefHeading___Toc120698944)

8.9.4.2.4 Type: RegistrationInformation [105](#__RefHeading___Toc120698945)

8.9.4.3 Simple data types and enumerations [105](#__RefHeading___Toc120698946)

8.9.4.3.1 Introduction [105](#__RefHeading___Toc120698947)

8.9.4.3.2 Simple data types [105](#__RefHeading___Toc120698948)

8.9.4.3.3 Enumeration: ApiProviderFuncRole [106](#__RefHeading___Toc120698949)

8.9.5 Error Handling [106](#__RefHeading___Toc120698950)

8.9.6 Feature negotiation [106](#__RefHeading___Toc120698951)

8.10 CAPIF\_Routing\_Info\_API [106](#__RefHeading___Toc120698952)

8.10.1 API URI [106](#__RefHeading___Toc120698953)

8.10.2 Resources [106](#__RefHeading___Toc120698954)

8.10.2.1 Overview [106](#__RefHeading___Toc120698955)

8.10.2.2 Resource: Individual Service API routing info [107](#__RefHeading___Toc120698956)

8.10.2.2.1 Description [107](#__RefHeading___Toc120698957)

8.10.2.2.2 Resource Definition [107](#__RefHeading___Toc120698958)

8.10.2.2.3 Resource Standard Methods [107](#__RefHeading___Toc120698959)

8.10.2.2.3.1 GET [107](#__RefHeading___Toc120698960)

8.10.2.2.4 Resource Custom Operations [108](#__RefHeading___Toc120698961)

8.10.3 Notifications [108](#__RefHeading___Toc120698962)

8.10.4 Data Model [108](#__RefHeading___Toc120698963)

8.10.4.1 General [108](#__RefHeading___Toc120698964)

8.10.4.2 Structured data types [108](#__RefHeading___Toc120698965)

8.10.4.2.1 Introduction [108](#__RefHeading___Toc120698966)

8.10.4.2.2 Type: RoutingInfo [108](#__RefHeading___Toc120698967)

8.10.4.2.3 Type: RoutingRule [109](#__RefHeading___Toc120698968)

8.10.4.2.4 Type: Ipv6AddressRange [109](#__RefHeading___Toc120698969)

8.10.4.3 Simple data types and enumerations [109](#__RefHeading___Toc120698970)

8.10.5 Error Handling [109](#__RefHeading___Toc120698971)

8.10.6 Feature negotiation [109](#__RefHeading___Toc120698972)

9 AEF API Definition [109](#__RefHeading___Toc120698973)

9.1 AEF\_Security\_API [109](#__RefHeading___Toc120698974)

9.1.1 API URI [109](#__RefHeading___Toc120698975)

9.1.2 Resources [110](#__RefHeading___Toc120698976)

9.1.2.2.4 Resource Custom Operations [110](#__RefHeading___Toc120698977)

9.1.2a Custom Operations without associated resources [110](#__RefHeading___Toc120698978)

9.1.2a.1 Overview [110](#__RefHeading___Toc120698979)

9.1.2a.2 Operation: check-authentication [110](#__RefHeading___Toc120698980)

9.1.2a.2.1 Description [110](#__RefHeading___Toc120698981)

9.1.2a.2.2 Operation Definition [110](#__RefHeading___Toc120698982)

9.1.2a.3 Operation: revoke-authorization [111](#__RefHeading___Toc120698983)

9.1.2a.3.1 Description [111](#__RefHeading___Toc120698984)

9.1.2a.3.2 Operation Definition [111](#__RefHeading___Toc120698985)

9.1.3 Notifications [111](#__RefHeading___Toc120698986)

9.1.4 Data Model [111](#__RefHeading___Toc120698987)

9.1.4.1 General [111](#__RefHeading___Toc120698988)

9.1.4.2 Structured data types [112](#__RefHeading___Toc120698989)

9.1.4.2.1 Introduction [112](#__RefHeading___Toc120698990)

9.1.4.2.2 Type: CheckAuthenticationReq [112](#__RefHeading___Toc120698991)

9.1.4.2.3 Type: CheckAuthenticationRsp [112](#__RefHeading___Toc120698992)

9.1.4.2.4 Type: RevokeAuthorizationReq [112](#__RefHeading___Toc120698993)

9.1.4.2.5 Type: RevokeAuthorizationRsp [112](#__RefHeading___Toc120698994)

9.1.4.3 Simple data types and enumerations [113](#__RefHeading___Toc120698995)

9.1.5 Error Handling [113](#__RefHeading___Toc120698996)

9.1.6 Feature negotiation [113](#__RefHeading___Toc120698997)

10 Security [113](#__RefHeading___Toc120698998)

10.1 General [113](#__RefHeading___Toc120698999)

10.2 CAPIF-1/1e security [113](#__RefHeading___Toc120699000)

10.3 CAPIF-2/2e security and securely invoking service APIs [113](#__RefHeading___Toc120699001)

Annex A (normative): OpenAPI specification [114](#__RefHeading___Toc120699002)

A.1 General [114](#__RefHeading___Toc120699003)

A.2 CAPIF\_Discover\_Service\_API [114](#__RefHeading___Toc120699004)

A.3 CAPIF\_Publish\_Service\_API [115](#__RefHeading___Toc120699005)

A.4 CAPIF\_Events\_API [122](#__RefHeading___Toc120699006)

A.5 CAPIF\_API\_Invoker\_Management\_API [126](#__RefHeading___Toc120699007)

A.6 CAPIF\_Security\_API [130](#__RefHeading___Toc120699008)

A.7 CAPIF\_Access\_Control\_Policy\_API [136](#__RefHeading___Toc120699009)

A.8 CAPIF\_Logging\_API\_Invocation\_API [137](#__RefHeading___Toc120699010)

A.9 CAPIF\_Auditing\_API [139](#__RefHeading___Toc120699011)

A.10 AEF\_Security\_API [141](#__RefHeading___Toc120699012)

A.11 CAPIF\_API\_Provider\_Management\_API [143](#__RefHeading___Toc120699013)

A.12 CAPIF\_Routing\_Info\_API [146](#__RefHeading___Toc120699014)

Annex B (informative): Change history [148](#__RefHeading___Toc120699015)

# 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 8259: "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".

[19] 3GPP TS 29.571: "5G System; Common Data Types for Service Based Interfaces Stage 3".

[20] IETF RFC 7239: "Forwarded HTTP Extension".

[21] 3GPP TS 29.500: "5G System; Technical Realization of Service Based Architecture; Stage 3".[22] W3C HTML 4.01 Specification, <https://www.w3.org/TR/2018/SPSD-html401-20180327/>.

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

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

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

[26] 3GPP TS 29.523: "5G System; Policy Control Event Exposure Service; Stage 3".

[27] 3GPP TR 21.900: "Technical Specification Group working methods".

[28] 3GPP TS 29.510: "5G System; Network Function Repository Services; Stage 3"

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

# 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 [2], clause 6 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).

3GPP TS 23.222 [2], clause 6 specifies functional model for the CAPIF to support 3rd party API providers, which is depicted in Figure 4.2-2 in detail



Figure 4.2-2: Functional model for the CAPIF to support 3rd party API providers

The CAPIF core function in the PLMN trust domain supports service APIs from both the PLMN trust domain and the 3rd party trust domain having business relationship with PLMN Trust Domain. The API invokers may exist within the PLMN trust domain, or within the 3rd party trust domain or outside of both the PLMN trust domain and the 3rd party trust domain.

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

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

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

CAPIF-7 and CAPIF-7e reference points connect API exposing functions within PLMN Trust Domain and outside PLMN Trust Domains respectively. 3GPP TS 23.222 [2] specifies functional model for interactions between API exposing functions.

NOTE: CAPIF-7 reference point is not represented in the Figure 4.2-2 which is aligning to TS 23.222 [2].

3GPP TS 23.222 [2], clause 6 specifies functional model to support CAPIF interconnection, which is depicted in Figure 4.2-3 in detail.



Figure 4.2-3: CAPIF interconnection functional model

CAPIF-6 and CAPIF-6e reference points connect two CAPIF core functions located in the same or different PLMN trust domains, respectively. The reference points allows API invokers of a CAPIF provider to utilize the service APIs from the 3rd party CAPIF provider or another CAPIF provider within trust domain.

## 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 and CAPIF-3e reference points 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;

- providing API routing information;

- providing API topology hiding information;

- logging of service API invocations; and

- charging of service API invocations.

The CAPIF core function supports the following capabilities over CAPIF-4 and CAPIF-4e reference points 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 and CAPIF-5e reference points 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.

- registering of API provider domain functions.

- update registration information of API provider domain functions.

The CAPIF core function supports the following capabilities over CAPIF-6 and CAPIF-6e reference point as defined in 3GPP TS 23.222 [2]:

- publishing of service APIs information with another CAPIF core function; and

- discovery of service APIs information with another CAPIF core function.

### 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 and CAPIF-3e reference points 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 communicates with the destination API exposing function over CAPIF-7/7e reference points.

### 4.3.4 API publishing function

The API publishing function (APF) enables the API provider to publish the Service APIs information using CAPIF-4 and CAPIF-4e reference points 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, register and maintain registration information of the API provider domain functions on the CAPIF core function, using CAPIF-5 and CAPIF-5e reference points as defined 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 | API Invoker, CAPIF core function |
| Event operations (NOTE) | (NOTE) | API Invoker |
| CAPIF\_Publish\_Service\_API | Publish\_Service\_API | Request/ Response | API Publishing Function, CAPIF core function |
| Unpublish\_Service\_API | Request/ Response | API Publishing Function, CAPIF core function |
| Update\_Service\_API | Request/ Response | API Publishing Function, CAPIF core function |
| Get\_Service\_API | Request/ Response | API Publishing Function, CAPIF core function |
| CAPIF\_Events\_API | Subscribe\_Event | Subscribe/Notify | 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 | Subscribe/Notify | 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 |
| Update\_API\_Invoker\_Details | Request/Response | API Invoker |
| Notify\_Update\_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 |
| CAPIF\_API\_Provider\_Management\_API | Register\_API\_Provider | Request/Response | API Management Function |
| Update\_API\_Provider | Request/Response | API Management Function |
| Deregister\_API\_Provider | Request/Response | API Management Function |
| CAPIF\_Routing\_Info\_API | Obtain\_ Routing\_Info | 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. | | | |

Table 5.1-2 summarizes the corresponding APIs defined in this specification.

Table 5.1-2: API Descriptions

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Service Name** | **Subclause** | **Description** | **OpenAPI Specification File** | **apiName** | **Annex** |
| CAPIF\_Discover\_Service\_API | 8.1 | CAPIF API discovery service | TS29222\_CAPIF\_Discover\_Service\_API.yaml | service-apis | A.2 |
| CAPIF\_Publish\_Service\_API | 8.2 | CAPIF API Publish Service | TS29222\_CAPIF\_Publish\_Service\_API.yaml | published-apis | A.3 |
| CAPIF\_Events\_API | 8.3 | CAPIF event service | TS29222\_CAPIF\_Events\_API.yaml | capif-events | A.4 |
| CAPIF\_API\_Invoker\_Management\_API | 8.4 | CAPIF API Invoker Management Service | TS29222\_CAPIF\_API\_Invoker\_Management\_API.yaml | api-invoker-management | A.5 |
| CAPIF\_Security\_API | 8.5 | CAPIF Security Service | TS29222\_CAPIF\_Security\_API.yaml | capif-security | A.6 |
| CAPIF\_Access\_Control\_Policy\_API | 8.6 | CAPIF Access Control Policy API Service | TS29222\_CAPIF\_Access\_Control\_Policy\_API.yaml | access-control-policy | A.7 |
| CAPIF\_Logging\_API\_Invocation\_API | 8.7 | CAPIF Logging API Invocation Service | TS29222\_CAPIF\_Logging\_API\_Invocation\_API.yaml | api-invocation-logs | A.8 |
| CAPIF\_Auditing\_API | 8.8 | CAPIF Auditing API Service | TS29222\_CAPIF\_Auditing\_API.yaml | logs | A.9 |
| CAPIF\_API\_Provider\_Management\_API | 8.9 | CAPIF API Provider Management API Service | TS29222\_CAPIF\_API\_Provider\_Management\_API.yaml | api-provider-management | A.11 |
| CAPIF\_Routing\_Info\_API | 8.10 | CAPIF Routing Information API Service | TS29222\_CAPIF\_Routing\_Info\_API.yaml | capif-routing-info | A.12 |

## 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-1/1e reference points to discover service API available at the CAPIF core function, and allow CAPIF core function via CAPIF-6 and CAPIF-6e reference points to discover service API available at other 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. This service operation is also used by CAPIF core function to discover service APIs available at other CAPIF core function. | API invoker, CAPIF core function |

#### 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; or

- a CAPIF core function to discover service API available at other CAPIF core function in interconnection scenario.

##### 5.2.2.2.2 Consumer discovering service API using Discover\_Service\_API service operation

To discover service APIs available at the CAPIF core function, the consumer (e.g. API invoker) shall send an HTTP GET message with the API invoker Identifier or CAPIF core function Identifier and query parameters to the CAPIF core function as specified in subclause 8.1.2.2.3.1.

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

1. verify the identity of the consumer (e.g. API invoker) and check if the consumer is authorized to discover the service APIs;

2. if the consumer 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 to obtain the list of service API description or the information of the CAPIF core function which is required to be contacted further for discovering the service APIs;

c. return the filtered search results or the information of the CAPIF core function in the response message. The shareableInformation for each of serviceAPIDescription is not provided in the filtered search results.

## 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 and CAPIF-4e reference points to publish and manage published service APIs at the CAPIF core function, and allow CAPIF core function via CAPIF-6 and CAPIF-6e reference points to publish and manage published service APIs at other CAPIF core function.

NOTE: Functions from 3rd party API provider domain can also access this API with sufficient permissions.

### 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. This service operation is also used by CAPIF core function to publish service APIs on other CAPIF core function. | API publishing function, CAPIF core function |
| Unpublish\_Service\_API | This service operation is used by an API publishing function to un-publish service APIs from the CAPIF core function. This service operation is also used by CAPIF core function to un-publish service APIs on other CAPIF core function. | API publishing function, CAPIF core function |
| Get\_Service\_API | This service operation is used by an API publishing function to retrieve service APIs from the CAPIF core function. This service operation is also used by CAPIF core function to retrieve service APIs on other CAPIF core function. | API publishing function, CAPIF core function |
| Update\_Service\_API | This service operation is used by an API publishing function to update published service APIs on the CAPIF core function. This service operation is also used by CAPIF core function to update published service APIs on other CAPIF core function. | API publishing function, CAPIF core 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: or

- a CAPIF core function to publish service APIs on other CAPIF core function in interconnection scenario.

##### 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 Information as specified in subclause 8.2.2.2.3.1.

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. If topology hiding is enabled as per policy, the CAPIF core function shall

i. determine the service APIs which require topology hiding as per policy;

ii. determine the API exposing function(s) responsible for the topology hiding for each service API which requires topology hiding;

iii. create a API topology hiding information for each service API which requires topology hiding by extracting the API identification information and the API exposing function(s) information from the service API information added to the CAPIF core function (API registry);

iv. replace the API exposing function(s) information in the service API information added to the CAPIF core function (API registry) with the corresponding API exposing function(s) information responsible for the topology hiding for service API;

v. send a notification message with the API topology hiding information to the API exposing function(s) which is responsible for the topology hiding for a service API and that has subscribed to the API\_TOPOLOGY\_HIDING\_CREATED event;

vi. store the API topology hiding information in the CAPIF core function;

c. create a new resource using the service API information in the CAPIF core function (API registry) as specified in subclause 8.2.2.1;

d. send a notification message with the updated service API, to all API Invokers that subscribed to the Service API Update event; and

e. return the CAPIF Resource URI in the response message.

##### 5.3.2.2.3 CAPIF core function publishing service APIs on other CAPIF core function using Publish\_Service\_API service operation

To publish service APIs at other CAPIF core function, the CAPIF core function shall send an HTTP POST message to the peer CAPIF core function. The body of the HTTP POST message shall include API Information as specified in subclause 8.2.2.2.3.1. For service API publishing on CAPIF-6 reference point, The CAPIF core function shall also include the published API path "pubApiPath" as specified in subclause 8.2.4.2.2. The "pubApiPath" includes a list of CAPIF core function Identifiers within the same CAPIF provider domain, such list includes own CAPIF core function identifier and received CAPIF core function identifier(s) from other CAPIF core function.

If the CAPIF core function knows the peer CAPIF core function identifier, it shall not send the HTTP POST message to the peer CAPIF core function if the peer CAPIF core function identifier is included in the published API path.

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

1. verify the identity of the requesting CAPIF core function in the URI and check if the CAPIF core function is authorized to publish service APIs;

2. if the requesting CAPIF core function is authorized to publish service APIs, the CAPIF core function shall check if own CAPIF core function identifier is within the published API path (if received). If it is not within the path, the CAPIF core function shall add its own identifier in the path; otherwise reject the HTTP POST request.

3. Then the CAPIF core function shall:

a. verify the rest 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 specified in subclause 8.2.2.1;

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; or

- a CAPIF core function to un-publish service APIs on other CAPIF core function in interconnection scenario.

##### 5.3.2.3.2 Consumer 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 consumer (e.g. API publishing function) shall send an HTTP DELETE message using the CAPIF Resource URI received during the publish operation to the CAPIF core function as specified in subclause 8.2.2.3.3.3.

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

1. verify the identity of the consumer (e.g. API publishing function) and check if the consumer is authorized to un-publish service APIs;

2. if the consumer 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);

c. If topology hiding is enabled as per policy, the CAPIF core function shall

i. determine the API topology hiding information associated with the service API and delete the corresponding API topology hiding information in the CAPIF core function;

ii. send a notification message with the deleted API topology hiding information to the corresponding API exposing function(s) which were responsible for the topology hiding of the service API and that subscribed to the API\_TOPOLOGY\_HIDING\_REVOKED event; and

d. 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; or

- a CAPIF core function to retrieve service APIs from other CAPIF core function in interconnection scenario.

##### 5.3.2.4.2 Consumer 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 consumer (e.g. API publishing function) shall send an HTTP GET message to the CAPIF core function. For retrieving the entire list of service APIs, the HTTP GET message shall be sent to the collection of service APIs resource representation URI as specified in subclause 8.2.2.2.3.2. For retrieving a specific service API, the HTTP GET message shall be sent to that service API’s resource representation URI as described in subclause 8.2.2.3.3.1.

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

1. verify the identity of the consumer (e.g. API publishing function) and check if the consumer is authorized to retrieve information about the published service APIs;

2. if the consumer is authorized to retrieve information about the published service APIs, the CAPIF core function shall:

a. respond with the requested API Information.

#### 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; or

- a CAPIF core function to update published service APIs on other CAPIF core function in interconnection scenario.

##### 5.3.2.5.2 Consumer updating published service APIs on CAPIF core function using Update\_Service\_API service operation

To update information of published service APIs, the consumer (e.g. API publishing function) shall send an HTTP PUT message to that service API’s resource representation URI in the CAPIF core function. The body of the HTTP PUT message shall include updated API Information as specified in subclause 8.2.2.3.3.2.

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

1. verify the identity of the consumer (e.g. API publishing function) and check if the consumer is authorized to update information of published service APIs;

2. if the consumer 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. If topology hiding is enabled as per policy, the CAPIF core function shall

i. if the service API being updated has a corresponding API topology hiding information in the CAPIF core function, then update the API topology hiding information with any updated API exposing function(s) information from the service API information replaced at the CAPIF core function (API registry);

ii. replace the API exposing function(s) information in the service API information added to the CAPIF core function (API registry) with the corresponding API exposing function(s) information responsible for the topology hiding for service API;

iii. send a notification message with the API topology hiding information to the API exposing function(s) which is responsible for the topology hiding for a service API and that has subscribed to the API\_TOPOLOGY\_HIDING\_CREATED event;

iv. update the API topology hiding information in the CAPIF core function;

c. replace the existing resource accordingly using the updated service API information in the CAPIF core function (API registry); and

d. 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-1/1e reference points, API exposure function via CAPIF-3/3e reference points, API publishing function via CAPIF-4/4e reference points and API management function via CAPIF-5/5e reference points 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 as specified in subclause 8.3.2.2.3.1.

For all events included in the HTTP POST message, if the Enhanced\_event\_report feature is supported, the Subscriber may include an event report requirement in the "eventReq" attribute including:

- event notification method (periodic, one time, on event detection) in the "notifMethod" attribute;

- maximum Number of Reports in the "maxReportNbr" attribute;

- monitoring duration in the "monDur" attribute;

- repetition period for periodic reporting in the "repPeriod" attribute; and/or

- immediate reporting indication in the "immRep" attribute.

If the Enhanced\_event\_report feature is supported, the Subscriber may also include an event filter in the "eventFilters" attribute. The "eventFilters" attribute shall include:

- if the event is SERVICE\_API\_AVAILABLE, SERVICE\_API\_UNAVAILABLE or SERVICE\_API\_UPDATE, the API IDs in the "apiIds" attribute;

- if the event is API\_INVOKER\_ONBOARDED or API\_INVOKER\_OFFBOARDED or API\_INVOKER\_UPDATED, the API invoker IDs in the "apiInvokerIds" attribute;

- if the event is ACCESS\_CONTROL\_POLICY\_UPDATE, the API invoker IDs in the "apiInvokerIds" attribute and/or API identifications in the "apiIds" attribute; and/or

- if the event is SERVICE\_API\_INVOCATION\_SUCCESS or SERVICE\_API\_INVOCATION\_FAILURE, the API invoker IDs in the "apiInvokerIds" attribute, AEF identifiers in the "aefIds" attribute and/or API IDs in the "apiIds" attribute.

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 specified in subclause 8.3.2.1; 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 to the resource representing the event in the CAPIF core function as specified in subclause 8.3.2.3.3.1.

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.

If the Enhanced\_event\_report feature is supported, the CAPIF core function may include an event detail in the "eventDetail" attribute. The "eventDetail" attribute shall include:

- if the event is SERVICE\_API\_AVAILABLE or SERVICE\_API\_UNAVAILABLE, the API IDs in the "apiIds" attribute;

- if the event is SERVICE\_API\_UPDATE, the API descriptions in the "serviceAPIDescriptions" attribute;

- if the event is API\_INVOKER\_ONBOARDED or API\_INVOKER\_OFFBOARDED or API\_INVOKER\_UPDATED, the API invoker IDs in the "apiInvokerIds" attribute;

- if the event is ACCESS\_CONTROL\_POLICY\_UPDATE, the access control policy information in the "accCtrlPolList" attribute;

- if the event is SERVICE\_API\_INVOCATION\_SUCCESS or SERVICE\_API\_INVOCATION\_FAILURE, the API invocation logs in the "invocationLogs" attribute; or

- if the event is API\_TOPOLOGY\_HIDING\_CREATED or API\_TOPOLOGY\_HIDING\_REVOKED, the API topology hiding information in the "apiTopoHide" attribute.

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-1/1e reference points to on-board and off-board itself as a recognized user of the CAPIF or update the API invoker’s details on the CAPIF core function.

### 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 |
| Update\_API\_Invoker\_Details | This service operation is used by an API invoker to update API invoker’s details in the CAPIF core function. | API Invoker |
| Notify\_Update\_Completion | This service operation is used by CAPIF core function to send an update 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 as specified in subclause 8.4.2.2.3.1.

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 8.4.2.1;

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 to its resource representation in the CAPIF core function as specified in subclause 8.4.2.3.3.1.

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.

#### 5.5.2.5 Update\_API\_Invoker\_Details

##### 5.5.2.5.1 General

This service operation is used by an API invoker to update the API invoker’s profile details on the CAPIF core function.

##### 5.5.2.5.2 API invoker updating its details on CAPIF using Update\_API\_Invoker\_Details service operation

To update the API invoker’s profile details on the CAPIF core function, the API invoker shall send a HTTP PUT message to the CAPIF core function to its resource representation, requesting to replace all properties in the existing resource, addressed by the URI received in the response to the request that has created the API invoker profile resource. The properties "apiInvokerId” and “onboardingInformation” shall remain unchanged from the previously provided values. The body of the HTTP PUT message shall include API invoker identity information, API invoker details that need to be updated and a Notification Destination URI for update notification as specified in clause 8.4.2.3.3.2. API invoker details may include API invoker information and API List. Upon receiving the above described HTTP PUT message:

1. if the CAPIF core function decides to update the API list of the API invoker without validation by CAPIF administrator, then the CAPIF core function:

a. shall determine if the request in the HTTP PUT message by the API invoker is authorized or not.

b. verify that the "apiInvokerId” and “onboardingInformation” properties are same as in API invoker resource on CAPIF core function.

c. if the API invoker's request is authorized and the properties "apiInvokerId” and “onboardingInformation” match, the CAPIF core function shall:

i. if the request contains an API list:

- create a list of APIs the API invoker is allowed to access; and

- update the resource identified by the CAPIF Resource URI of the API invoker’s HTTP PUT request with the updated information in the request and the API list created in step A;

ii. if the request does not contain an API list, update the resource identified by the CAPIF Resource URI of the API invoker’s HTTP PUT request with the updated information in the request; and

iii. return the updated API invoker details;

2. otherwise, the CAPIF core function:

a. shall acknowledge the receipt of the update API invoker details request to the API invoker.

b. verify that the "apiInvokerId” and “onboardingInformation” properties are same as in API invoker resource on CAPIF core function;

c. if the properties "apiInvokerId” and “onboardingInformation” match, then shall request the CAPIF administrator to validate the request or the API management to validate the request by sharing the API invoker identity information and the updated information received in the HTTP PUT message;

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

i. update the resource identified by the CAPIF Resource URI of the API invoker’s HTTP PUT request, with validated information; and

ii. return the updated API invoker details.

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

NOTE 2: Interactions between the CAPIF core function and the CAPIF administrator or the API management is out-of-scope of this specification.

NOTE 3: After the operation is completed the API Invoker no longer needs to maintain the Notification Destination URI and may delete it.

#### 5.5.2.6 Notify\_Update\_Completion

##### 5.5.2.6.1 General

This service operation is used by the CAPIF core function to send a notification about the completion of the update of API invoker’s details.

##### 5.5.2.6.2 Notifying API invoker update completion using Notify\_Update\_Completion service operation

When the CAPIF core function cannot immediately grant the update request (see subclause 5.5.2.5.2) 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 update details request. The body of the HTTP POST message shall include the updated API Invoker details.

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 update the details of the API invoker request, and respond to the HTTP POST message with HTTP response 204 No content.

## 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 (re-)negotiate the service security method and obtain authorization for invoking service APIs; and

- API exposing function via CAPIF-3/3e reference points to obtain authentication information of the API invoker for authentication of the API invoker and revoke the authorization for service APIs.

### 5.6.2 Service Operations

#### 5.6.2.1 Introduction

The service operations defined for CAPIF\_Security\_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 |

Security information is generated when requested by an API invoker, and is stored in the CAPIF Core function. The information can be accessed via a resource representation URI using the API invoker ID as described in subclause 8.5.2.3. The URI is provided to the API invoker in the HTTP response to the creation request (via the Obtain\_Security\_Method service operation name).

Refer to subclause 9.1.2a.2 for details about verifying that the API Exposing function has the ability to authorize API invokers prior to invoking service APIs.

#### 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 PUT message to the CAPIF core function. The body of the HTTP PUT 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 as specified in subclause 8.5.2.3.3.3.

Upon receiving the above described HTTP PUT 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 8.5.2.1; 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].

The API invoker shall send a POST request to the "Token Endpoint", as described in IETF RFC 6749 [23], subclause 3.2. The "Token Endpoint" URI shall be:

{apiRoot}/capif-security/v1/securities/{securityId}/token

where {securityId} is the API invoker identifier and represents the "Individual trusted API invoker" resource created during obtain security method, as described in subclause 5.6.2.2.  
  
The body of the HTTP POST request shall indicate that the required OAuth2 grant must be of type "client\_credentials". The "scope" parameter (if present) shall include a list of AEF identifiers and its associated API names the API invoker is trying to access (i.e., the API invoker expected scope).

The API invoker may use HTTP Basic authentication towards this endpoint, using the API invoker identifier as "username" and the onboarding secret as "password". Such username and password may be included in the header or body of the HTTP POST request.

On success, "200 OK" shall be returned. The payload body of the POST response shall contain the requested access token, the token type and the expiration time for the token. The access token shall be a JSON Web Token (JWT) as specified in IETF RFC 7519 [24]. The access token returned by the CAPIF core function shall include the claims encoded as a JSON object as specified in subclause 8.5.4.2.8 and then digitally signed using JWS as specified in IETF RFC 7515 [25] and in Annex C.1 of 3GPP TS 33.122 [16].

The digitally signed access token shall be converted to the JWS Compact Serialization encoding as a string as specified in subclause 7.1 of IETF RFC 7515 [25].

If the access token request fails at the CAPIF core function, the CAPIF core function shall return "400 Bad Request" status code, including a JSON object in the response payload, that includes details about the specific error that occurred.

#### 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 that API invoker’s resource representation URI in the CAPIF core function with an indication to request authentication and/or authorization information, as specified in subclause 8.5.2.3.3.1.

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.

NOTE: Functions from 3rd party API provider domain can also access this service operation with sufficient permissions.

#### 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 for all service APIs, the API exposing function shall send an HTTP DELETE message to that API invoker’s resource representation URI in the CAPIF core function using the API invoker ID as specified in subclause 8.5.2.3.3.2.

Upon receiving the HTTP DELETE message, the CAPIF core function shall delete the resource representation and shall notify the API invoker of the authorization invalidation using the Notification Destination URI received in the Obtain\_Security\_Method message.

The CAPIF core function shall also invalidate the previously assigned access token when the authorization of all service APIs are revoked for the API invoker.

To invalidate authorization of an API invoker for some service APIs, the API exposing function shall send an HTTP POST message to that API invoker’s "delete" custom resource representation URI in the CAPIF core function with a list of the service APIs that should be revoked.

Upon receiving the HTTP POST message, the CAPIF core function shall revoke the authorization of the API invoker for the indicated service APIs (e.g. it may update the list of unauthorized APIs locally); and shall notify the API invoker of the authorization invalidation using the Notification Destination URI received in the Obtain\_Security\_Method message.

In both alternatives, the CAPIF core function shall acknowledge the HTTP request from the API exposing function.

NOTE: Functions from 3rd party API provider domain can also access this service operation with sufficient permissions.

## 5.7 CAPIF\_Monitoring\_API

The CAPIF monitoring API as defined in 3GPP TS 23.222 [2], allow the API management function via CAPIF-5/5e reference points 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 one of the events as described in subclause 8.3.4.3.3.

## 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-3/3e reference points to log the information related to service API invocations on the CAPIF core function.

NOTE: Functions from 3rd party API provider domain can also access this API with sufficient permissions.

### 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 as specified in subclause 8.7.2.2.3.1.

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 8.7.2.1; 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-5/5e reference points to query the log information stored on the CAPIF core function.

NOTE: Functions from 3rd party API provider domain can also access this API with sufficient permissions.

### 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 as specified in subclause 8.8.2.2.3.1.

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-3/3e reference points to obtain the service API access policy from the CAPIF core function.

NOTE: Functions from 3rd party API provider domain can also access this API with sufficient permissions.

### 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 as specified in subclause 8.6.2.2.3.1.

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 8.3.4.3.3.

## 5.11 CAPIF\_API\_Provider\_Management\_API

### 5.11.1 Service Description

#### 5.11.1.1 Overview

The CAPIF API provider management APIs, as defined in 3GPP TS 23.222 [2], allow API management functions via CAPIF-5 and CAPIF-5e reference points to register, deregister and update registration information of API provider domain functions (API Exposing Function, API Publishing Function, API management Function) as a recognized API provider domain of the CAPIF domain.

### 5.11.2 Service Operations

#### 5.11.2.1 Introduction

The service operations defined for the CAPIF API Provider Management API are shown in table 5.11.2.1-1.

Table 5.11.2.1-1: Operations of the CAPIF\_API\_Provider\_Management\_API

| **Service operation name** | **Description** | **Initiated by** |
| --- | --- | --- |
| Register\_API\_Provider | This service operation is used by an API management function to register API provider domain functions as a recognized API provider domain of the CAPIF domain. | API Management Function |
| Update\_API\_Provider | This service operation is used by an API management function to update the API provider domain functions details in the CAPIF domain. | API Management Function |
| Deregister\_API\_Provider | This service operation is used by an API management function to deregister API provider domain functions as a recognized API provider domain of the CAPIF domain. | API Management Function |

#### 5.11.2.2 Register\_API\_Provider

##### 5.11.2.2.1 General

This service operation is used by an API management function to register API provider domain functions as a recognized API provider of CAPIF domain.

##### 5.11.2.2.2 API provider domain functions registering as a recognized API provider domain function of CAPIF using Register\_API\_Provider service operation

To register API provider domain as a recognized API provider of the CAPIF, the API management function shall send a HTTP POST message to the CAPIF core function. The body of the HTTP POST message shall include API provider Enrolment Details, consisting of details of all API provider domain functions and security information for CAPIF core function to validate the registration request.

Upon receiving the above described HTTP POST message, the CAPIF core function validates the security information and determine if the request sent by API management function is authorized or not. If the API management function is authorized, CAPIF core function shall:

a. create the API provider domain profile consisting of API provider domain ID, API provider domain functions profiles as per the request. CAPIF core function shall assign the identities for the API provider domain functions.;

b. create a new resource as defined in clause 8.9.2.2.3.1; and

c. return the API provider domain profile, the CAPIF Resource URI in the response message and registration failure information specific to individual API provider domain functions.

#### 5.11.2.3 Update\_API\_Provider

##### 5.11.2.3.1 General

This service operation is used by an API management function to update API provider domain function details on the CAPIF domain.

##### 5.11.2.3.2 API management function updating API provider domain function details on CAPIF using Update\_API\_Provider service operation

To update the API provider domain profile and its individual functions details on CAPIF domain, the API management function shall send a HTTP PUT message to its resource representation in the CAPIF core function as specified in clause 8.9.2.3.3.1, requesting to replace all properties in the existing resource, addressed by the URI received in the response to the request that has created the API provider domain profile resource. The property “apiProviderDomainId”, shall remain unchanged from the previously provided values. The body of the HTTP PUT message shall include the API provider domain profile that need to be updated. Upon receiving the described HTTP PUT message:

1. the CAPIF core function shall process the updates received in the HTTP PUT message and determine if the request sent by API management function is authorized or not;

2. verify that the “apiProviderDomainId” property is same as in the API provider domain resource on CAPIF Core Function.

3. if the API management function is authorized and the property “apiProviderDomainId” matches, then the CAPIF core function shall

a. replace the representation of the resource identified by the CAPIF Resource URI of the API management function’s HTTP PUT request with updated information in the request.

b. update the individual API provider domain function profiles as per the request. CAPIF core function shall create new API provider domain function profiles along with assignment of identities, if the API provider domain functions profiles in the request do not exist in CAPIF.

c. return the updated API provider domain profile resource and update failure information specific to individual API provider domain functions.

#### 5.11.2.4 Deregister\_API\_Provider

##### 5.11.2.4.1 General

This service operation is used by an API management function to deregister the API provider domain function as a recognized API provider of the CAPIF domain.

##### 5.11.2.4.2 API provider domain functions deregistering as a recognized API provider domain function of CAPIF using Deregister\_API\_Provider service operation

To deregister API provider domain as a recognized API provider of the CAPIF domain, the API management function shall send an HTTP DELETE message to its resource representation in the CAPIF core function as specified in clause 8.9.2.3.3.2.

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

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

2. if the API management function’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 provider domain profile.

## 5.12 CAPIF\_Routing\_Info\_API

### 5.12.1 Service Description

#### 5.12.1.1 Overview

The CAPIF routing info API allows an API exposing function via CAPIF-3/3e reference point to obtain the API routing information from the CAPIF core function.

NOTE: Functions from 3rd party API provider domain can also access this API routing information with sufficient permissions.

### 5.12.2 Service Operations

#### 5.12.2.1 Introduction

Table 5.12.2.1-1: Operations of the CAPIF\_Routing\_Info\_API

| Service operation name | Description | Initiated by |
| --- | --- | --- |
| Obtain\_Routing\_Info | This service operation is used by an API exposing function to obtain the API routing information from the CAPIF core function. | API exposing function |

#### 5.12.2.2 Obtain\_Routing\_Info

##### 5.12.2.2.1 General

This service operation is used by an API exposing function to obtain the API routing information from the CAPIF core function.

##### 5.12.2.2.2 API exposing function obtaining API routing information from the CAPIF core function using Obtain\_Routing\_Info service operation

To obtain the API routing information from the CAPIF core function, the API exposing function shall send an HTTP GET request message to the CAPIF core function with the API exposing function Identifier and API identification as specified in subclause 8.10.2.2.3.1.

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 API routing information corresponding to the API identification;

2. if the API exposing function is authorized to obtain the API routing information, the CAPIF core function shall respond with the API routing information corresponding to the API identification in the HTTP GET response message.

# 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\_Security\_API | Initiate\_Authentication | Request/ Response | API Invoker |
| Revoke\_Authorization | Request/ Response | CAPIF core function |

Table 6.1-2 summarizes the corresponding APIs defined in this specification.

Table 6.1-2: API Descriptions

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Service Name** | **Subclause** | **Description** | **OpenAPI Specification File** | **apiName** | **Annex** |
| AEF\_Security\_API | 9.1 | AEF Security API Service | TS29222\_AEF\_Security\_API.yaml | aef-security | A.10 |

## 6.2 AEF\_Security\_API

### 6.2.1 Service Description

#### 6.2.1.1 Overview

The AEF securityAPI, 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.

The AEF security API, also allows the CAPIF core function via CAPIF-3/3e reference points to request API exposing function to revoke the authorization of service APIs for an API invoker.

### 6.2.2 Service Operations

#### 6.2.2.1 Introduction

The service operation defined for AEF\_Security\_API is shown in table 6.2.2.1-1.

Table 6.2.2.1-1: Operations of the AEF\_Security\_API

| Service operation name | Description | Initiated by |
| --- | --- | --- |
| Initiate\_Authentication | This service operation is used by an API invoker to request API exposing function to confirm necessary authentication data is available to authenticate the API invoker | API invoker |
| Revoke\_Authorization | This service operation is used by the CAPIF core function to request the API exposing function to revoke the authorization of service APIs for an API invoker. | CAPIF core function |

#### 6.2.2.2 Initiate\_Authentication

##### 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 Initiate\_Authentication 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 Initiate\_Authentication service operation

To initiate authentication with the API exposing function, the API invoker shall send an HTTP POST message to the API exposing function with the API invoker ID to the URI "{apiRoot}/aef-security/v1/check-authentication".

Upon receiving the above described HTTP POST 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.

The API exposing function shall store the received credentials and respond to the API invoker with 200 OK status code.

#### 6.2.2.3 Revoke\_Authorization

##### 6.2.2.3.1 General

This service operation is used by CAPIF core function to revoke authorization of service APIs (e.g. due to policy change in the CAPIF core function). On receiving the Revoke\_Authorization the API exposing function revokes authorization of the API invoker for the service APIs indicated in the request.

##### 6.2.2.3.2 CAPIF core function initiating revocation using Revoke\_Authorization service operation

To revoke authorization, the CAPIF core function shall send an HTTP POST message to the API exposing function with the API invoker ID and a list of service API IDs on the URI "{apiRoot}/aef-security/v1/revoke-authorization".

Upon receiving the HTTP POST message, the API exposing function shall revoke the authorization of the API invoker for the indicated service APIs (e.g. it may update the list of unauthorized APIs locally), and then respond to the CAPIF core function with 200 OK status code.

The CAPIF core function shall also notify the API invoker of the authorization invalidation using the Notification Destination URI received in the Obtain\_Security\_Method message.

# 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 the Subscriber. |
| WebsockNotifConfig | 3GPP TS 29.122 [14] | Following clarifications apply:  - The SCEF is the CAPIF core function; and  - The SCS/AS is the Subscriber. |

### 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 8259 [12]).

The MIME media type that shall be used within the related Content-Type header field is "application/json", as defined in IETF RFC 8259 [12] unless specified otherwise in the API definition.

## 7.5 URI structure

7.5.1 Resource URI structure

All API URIs of CAPIF APIs shall be:

**{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 API 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 as resource URI where they apply:

**{apiRoot}/<apiName>/<apiVersion>/<apiSpecificSuffixes>**

7.5.2 Custom operations URI structure

The custom operation definition is in Annex C of 3GPP TS 29.501 [18].

The URI of a custom operation which is associated with a resource shall have the following structure:

**{apiRoot}/<apiName>/<apiVersion>/<apiSpecificResourceUriPart>/<custOpName>**

Custom operations can also be associated with the service instead of a resource. The URI of a custom operation which is not associated with a resource shall have the following structure:

**{apiRoot}/<apiName>/<apiVersion>/<custOpName>**

In the above URI structures, "apiRoot", "apiName", "apiVersion" and "apiSpecificResourceUriPart" are as defined in subclause 7.5.1 and "custOpName" represents the name of the custom operation as defined in subclause 5.1.3.2 of 3GPP TS 29.501 [18].

## 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 the Subscriber.

## 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:

- description of the SCEF applies to the CAPIF core function;

- description of the SCS/AS applies to the functional entity invoking an API;

- the CAPIF Core Function should not register any feature in the NRF; and

- the AEF should not register any feature for AEF\_Security\_API in the NRF.

## 7.9 HTTP headers

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

## 7.10 Conventions for Open API specification files

The conventions for Open API specification files as specified in subclause 5.2.9 of 3GPP TS 29.122 [14] shall be applicable for all APIs in this document.

# 8 CAPIF API Definition

## 8.1 CAPIF\_Discover\_Service\_API

### 8.1.1 API URI

The CAPIF\_Discover\_Service\_API service shall use the CAPIF\_Discover\_Service\_API.

The request URIs used in HTTP requests from the API invoker towards the CAPIF core function shall have the Resource URI 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) | /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/<apiVersion>/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 | Data Type | Definition |
| apiRoot | string | See clause 7.5 |
| apiVersion | string | See clause 8.1.1 |

##### 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 | Applicability |
| api-invoker-id | string | M | 1 | String identifying the API invoker assigned by the CAPIF core function. It also represents the CCF identifier in the CAPIF-6/6e reference point. |  |
| api-name | string | O | 0..1 | API name, it is set as {apiName} part of the URI structure as defined in subclause 4.4 of 3GPP TS 29.501 [18]. |  |
| api-version | string | O | 0..1 | API major version in the URI (e.g. v1) |  |
| comm-type | CommunicationType | O | 0..1 | Communication type used by the API (e.g.REQUEST\_RESPONSE). |  |
| protocol | Protocol | O | 0..1 | Protocol used by the API. |  |
| aef-id | string | O | 0..1 | AEF identifier. |  |
| data-format | DataFormat | O | 0..1 | Data format used by the API (e.g. serialization protocol JSON used). |  |
| api-cat | string | O | 0..1 | The service API category to which the service API belongs to. |  |
| supported-features | SupportedFeatures | O | 0..1 | To filter irrelevant responses related to unsupported features. |  |
| api-supported-features | SupportedFeatures | C | 0..1 | Features supported by the discovered service API indicated by api-name parameter. This may only be present if api-name query parameter is present. | ApiSupportedFeatureQuery |

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 |
| DiscoveredAPIs | M | 1 | 200 OK | The response body contains the result of the search over the list of registered APIs. |
| ProblemDetails | O | 0..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 |
| CommunicationType | Subclause 8.2.4.3.5 | Communication type used by the API |  |
| DataFormat | Subclause 8.2.4.3.4 | Data format |  |
| ProblemDetails | 3GPP TS 29.122 [14] |  |  |
| Protocol | Subclause 8.2.4.3.3 | Protocol |  |
| ServiceAPIDescription | Subclause 8.2.4.2.2 | Description of the service API |  |

#### 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 |
| serviceAPIDescriptions | array(ServiceAPIDescription) | O | 1..N | Description of the service API as published by the service. (NOTE) |  |
| NOTE: For CAPIF\_Discover\_Service\_API, the supportedFeatures attribute in ServiceAPIDescription data type shall be provided in the HTTP GET response of successful query. In addition, the supportedFeatures attribute may include one or more the supported features as defined in subclause 8.1.6. | | | | | |

##### 8.1.4.2.3 Void

#### 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** |
| 1 | ApiSupportedFeatureQuery | Indicates the support of query with supported feature for a service API. |

## 8.2 CAPIF\_Publish\_Service\_API

### 8.2.1 API URI

The CAPIF\_Publish\_Service\_API service shall use the CAPIF\_Publish\_Service\_API.

The request URIs used in HTTP requests from the API publishing function towards the CAPIF core function shall have the Resource URI 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 | /{apfId}/service-apis | POST | Publish a new API |
| GET | Retrieve all published service APIs |
| Individual APF published API | /{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/<apiVersion>/{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 | Data Type | Definition |
| apiRoot | string | See clause 7.5 |
| apiVersion | string | See clause 8.2.1 |
| apfId | string | Identifies the API publishing function; for CAPIF interconnection case, this string also identifies the CCF which is publishing the service API. |

##### 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 |

Table 8.2.2.2.3.1-4: Headers supported by the 201 Response Code on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data type | P | Cardinality | Description |
| Location | string | M | 1 | Contains the URI of the newly created resource, according to the structure: {apiRoot}/published-apis/<apiVersion>/{apfId}/service-apis/{serviceApiId} |

###### 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/<apiVersion>/{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 | Data Type | Definition |
| apiRoot | string | See clause 7.5 |
| apiVersion | string | See clause 8.2.1 |
| apfId | string | Identifies the API publishing function |
| serviceApiId | string | Identifies 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 |
| ServiceAPIDescription | M | 1 | 200 OK | Definition of individual 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 |
| AefProfile | 8.2.4.2.4 | AEF profile |  |
| CommunicationType | 8.2.4.3.5 | Communication type of the resource |  |
| CustomOperation | 8.2.4.2.7 | Custom operation |  |
| DataFormat | 8.2.4.3.4 | Data format |  |
| InterfaceDescription | 8.2.4.2.3 | Description of the API interface |  |
| Operation | 8.2.4.3.7 | HTTP method (e.g. PUT) |  |
| Protocol | 8.2.4.3.3 | Protocol used by the API |  |
| Resource | 8.2.4.2.6 | API resource |  |
| SecurityMethod | 8.2.4.3.6 | Security method (e.g. PKI) |  |
| Version | 8.2.4.2.5 | API version information |  |

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] |  |  |
| Port | 3GPP TS 29.122 [14] |  |  |
| SupportedFeatures | 3GPP TS 29.571 [19] | Used to negotiate the applicability of optional features defined in table  8.2.6-1. |  |

#### 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, it is set as {apiName} part of the URI structure as defined in subclause 4.4 of 3GPP TS 29.501 [18]. |  |
| apiId | string | O | 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 and in the HTTP GET response from the CAPIF core function to the API invoker (discovery API). |  |
| aefProfiles | array(AefProfile) | C | 1..N | AEF profile information, which includes the exposed API details (e.g. protocol). For CAPIF-4/4e interface, API publishing function shall provide this attribute to the CAPIF core function in service API publishing. For CAPIF-1/1e interface, the CAPIF core function shall provide this attribute to the API Invoker during service API discovery. (NOTE 2) |  |
| description | string | O | 0..1 | Text description of the API |  |
| supportedFeatures | SupportedFeatures | O | 0..1 | The supported optional features of the CAPIF API. (NOTE 1) |  |
| shareableInfo | ShareableInformation | O | 0..1 | Represents whether the service API and/or the service API category can be published to other CCFs. |  |
| serviceAPICategory | string | C | 0..1 | The service API category to which the service API belongs to. This attribute is only applicable for CAPIF-6/6e interface. (NOTE 2) |  |
| ccfId | string | C | 0..1 | CAPIF core function identifier which can be contacted further for discovering the details of service API information. This attribute is only applicable for CAPIF-6/6e interface and shall be provided with serviceAPICategory. (NOTE 2) |  |
| apiSuppFeats | SupportedFeatures | O | 0..1 | Provided by the consumer to indicate the features supported by the service API. | ApiSupportedFeaturePublishing |
| pubApiPath | PublishedApiPath | C | 0..1 | It contains the published API path within the same CAPIF provider domain. it shall be provided by the CCF when publishing the service API to other CCF via the CAPIF-6 reference point. |  |
| NOTE 1: For CAPIF\_Publish\_Service\_API, the supported features attribute shall be provided in the HTTP POST request and in the response of successful resource creation. In addition, the supportedFeatures attribute may include one or more the supported features as defined in subclause 8.2.6.  NOTE 2: For CAPIF-6/6e interface, at least one of aefProfiles or serviceAPICategory and the corresponding ccfId shall be provided. | | | | | |

##### 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 (NOTE) |  |
| ipv6Addr | Ipv6Addr | O | 0..1 | String identifying an IPv6 address (NOTE) |  |
| port | Port | O | 0..1 | Port |  |
| securityMethods | array(SecurityMethod) | M | 1..N | Security methods supported by the interface. It takes precedence over the security methods provided in AefProfile, for this specific interface |  |
| NOTE: Only one of the attributes "ipv4Addr" or "ipv6Addr" shall be included. | | | | | |

##### 8.2.4.2.4 Type: AefProfile

Table 8.2.4.2.4-1: Definition of type AefProfile

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description | Applicability |
| aefId | string | M | 1 | AEF identifier |  |
| versions | array(Version) | M | 1..N | API version |  |
| protocol | Protocol | O | 0..1 | Protocol used by the API. |  |
| dataFormat | DataFormat | O | 0..1 | Data format used by the API |  |
| securityMethods | array(SecurityMethod) | O | 1..N | Security methods supported by the AEF for all interfaces. Certain interfaces may have different security methods supported in the attribute interfaceDescriptions. |  |
| domainName | string | O | 0..1 | Domain to which API belongs to  (NOTE 1) |  |
| interfaceDescriptions | array(InterfaceDescription) | O | 1..N | Interface details  (NOTE 1) |  |
| NOTE 1: Only one of the attributes "domainName" or "interfaceDescriptions" shall be included.  NOTE 2: Notification or callback type of resource is not included. | | | | | |

##### 8.2.4.2.5 Type: Version

Table 8.2.4.2.5-1: Definition of type Version

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description | Applicability |
| apiVersion | string | M | 1 | API major version in URI (e.g. v1) |  |
| expiry | DateTime | O | 0..1 | Expiry date and time of the AEF service. This represents the planned retirement date as specified in subclause 4.3.1.5 of 3GPP TS 29.501 [18]. |  |
| resources | array(Resource) | O | 1..N | Resources supported by the API. It may include the custom operations with resource association. |  |
| custOperations | array(CustomOperation) | O | 1..N | Custom operations without resource association. |  |

##### 8.2.4.2.6 Type: Resource

Table 8.2.4.2.6-1: Definition of type Resource

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description | Applicability |
| resourceName | string | M | 1 | Resource name |  |
| commType | CommunicationType | M | 1 | Communication type used by the API resource |  |
| uri | string | M | 1 | Relative URI of the API resource, it is set as {apiSpecificResourceUriPart} part of the URI structure as defined in subclause 4.4 of 3GPP TS 29.501 [18]. |  |
| custOpName | string | O | 0..1 | it is set as {custOpName} part of the URI structure for a custom operation associated with a resource as defined in subclause 4.4 of 3GPP TS 29.501 [18]. |  |
| operations | array(Operation) | C | 1..N | Supported HTTP methods for the API resource. Only applicable when the protocol in AefProfile indicates HTTP. |  |
| description | string | O | 0..1 | Text description of the API resource. |  |

##### 8.2.4.2.7 Type: CustomOperation

Table 8.2.4.2.7-1: Definition of type CustomOperation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description | Applicability |
| commType | CommunicationType | M | 1 | Communication type used by the API resource |  |
| custOpName | string | M | 1 | it is set as {custOpName} part of the URI structure for a custom operation without resource association as defined in subclause 4.4 of 3GPP TS 29.501 [18]. |  |
| operations | array(Operation) | C | 1..N | Supported HTTP methods for the API resource. Only applicable when the protocol in AefProfile indicates HTTP. |  |
| description | string | O | 0..1 | Text description of the custom operation. |  |

##### 8.2.4.2.8 Type: ShareableInformation

Table 8.2.4.2.8-1: Definition of type ShareableInformation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description | Applicability |
| isShareable | boolean | M | 1 | Set to "true" indicates that the service API and/or the service API category can be shared to the list of CAPIF provider domain information. Otherwise set to "false". |  |
| capifProvDoms | array(string) | O | 1..N | List of CAPIF provider domains to which the service API information to be shared. (NOTE) |  |
| NOTE: Only one CAPIF provider domain information shall be provided via the CAPIF-6e interface. | | | | | |

##### 8.2.4.2.9 Type: PublishedApiPath

Table 8.2.4.2.9-1: Definition of type PublishedApiPath

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description | Applicability |
| ccfIds | array(string) | O | 1..N | A list of CCF identifiers where the service API is already published. |  |

#### 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 |  |
| HTTP\_2 | 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: SecurityMethod

Table 8.2.4.3.6-1: Enumeration SecurityMethod

|  |  |  |
| --- | --- | --- |
| 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]. |  |

##### 8.2.4.3.7 Enumeration: Operation

Table 8.2.4.3.7-1: Enumeration Operation

|  |  |  |
| --- | --- | --- |
| Enumeration value | Description | Applicability |
| GET | HTTP GET method |  |
| POST | HTTP POST method |  |
| PUT | HTTP PUT method |  |
| PATCH | HTTP PATCH method |  |
| DELETE | HTTP DELETE method |  |

### 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** |
| 1 | ApiSupportedFeaturePublishing | Indicates the support of publishing with supported feature for a service API. |

## 8.3 CAPIF\_Events\_API

### 8.3.1 API URI

The CAPIF\_Events\_API service shall use the CAPIF\_Events\_API.

The request URIs used in HTTP requests from the Subscriber towards the CAPIF core function shall have the Resource URI 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 | /{subscriberId}/subscriptions | POST | Creates a new individual CAPIF Event Subscription |
| Individual CAPIF Events Subscription | /{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/<apiVersion>/{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 | Data Type | Definition |
| apiRoot | string | See clause 7.5 |
| apiVersion | string | See clause 8.3.1 |
| subscriberId | string | 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 |

Table 8.3.2.2.3.1-4: Headers supported by the 201 Response Code on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data type | P | Cardinality | Description |
| Location | string | M | 1 | Contains the URI of the newly created resource, according to the structure: {apiRoot}/capif-events/<apiVersion>/{subscriberId}/subscriptions/{subscriptionId} |

##### 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/<apiVersion>/{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 | Data Type | Definition |
| apiRoot | string | See clause 7.5 |
| apiVersion | string | See clause 8.3.1 |
| subscriberId | string | ID of the Subscriber |
| subscriptionId | string | Identifies 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 | Callback 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 subscription to the event.

Callback 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 |
| AccessControlPolicyListExt | 8.3.4.2.6 | Represents the extension for access control policies |  |
| CAPIFEvent | 8.3.4.3.2 | Describes CAPIF events |  |
| CAPIFEventDetail | 8.3.4.2.5 | Represents the CAPIF event detail |  |
| CAPIFEventFilter | 8.3.4.2.4 | Represents the CAPIF event filter |  |
| EventNotification | 8.3.4.2.3 | Represents an individual CAPIF Event Subscription Notification |  |
| EventSubscription | 8.3.4.2.2 | Represents an individual CAPIF Event Subscription resource |  |
| TopologyHiding | 8.3.4.2.7 | Represents the routing rules information of a service API. |  |

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 |
| ReportingInformation | 3GPP TS 29.523 [26] | Used to indicate the reporting requirement, only the following information are applicable for CAPIF:  - immRep  - notifMethod  - maxReportNbr  - monDur  - repPeriod | Enhanced\_event\_report |
| SupportedFeatures | 3GPP TS 29.571 [19] | Used to negotiate the applicability of optional features defined in table 8.3.6-1. |  |
| 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. |  |
| Uri | 3GPP TS 29.122 [14] |  |  |
| 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 |  |
| eventFilters | array(CAPIFEventFilter) | O | 1..N | Subscribed event filters.  The nth entry in the "eventFilters" attribute shall correspond to the nth entry in the "events" attribute. For event not having event filter, an empty event filter entry without any sub-attribute shall be provided. | Enhanced\_event\_report |
| eventReq | ReportingInformation | O | 0..1 | Represents the reporting requirements of the event subscription. | Enhanced\_event\_report |
| 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 |
| supportedFeatures | SupportedFeatures | O | 0..1 | Used to negotiate the supported optional features of the API as described in subclause 7.8.  This attribute shall be provided in the HTTP POST request and in the response of successful resource creation. |  |

##### 8.3.4.2.3 Type: EventNotification

Table 8.3.4.2.3-1: 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 | CAPIFEvent | M | 1 | Notifications of individual events |  |
| eventDetail | CAPIFEventDetail | O | 0..1 | Detailed information for the event. | Enhanced\_event\_report |

##### 8.3.4.2.4 Type: CAPIFEventFilter

Table 8.3.4.2.4-1: Definition of type CAPIFEventFilter

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description | Applicability |
| apiIds | array(string) | O | 1..N | API identifiers that the event subscriber wants to know in the interested event. |  |
| apiInvokerIds | array(string) | O | 1..N | API invokers that the event subscriber wants to know in the interested event. |  |
| aefIds | array(string) | O | 1..N | String identifying the AEF. |  |

##### 8.3.4.2.5 Type: CAPIFEventDetail

Table 8.3.4.2.5-1: Definition of type CAPIFEventDetail

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description | Applicability |
| serviceAPIDescriptions | array(ServiceAPIDescription) | O | 1..N | Description of the service API as published by the APF. |  |
| apiIds | array(string) | O | 1..N | API identifiers. |  |
| apiInvokerIds | array(string) | O | 1..N | API invokers that are onboarded/offboarded. |  |
| accCtrlPolList | AccessControlPolicyListExt | O | 0..1 | Access control policy updated list. |  |
| invocationLogs | array(InvocationLog) | O | 1..N | Invocation logs |  |
| apiTopoHide | TopologyHiding | O | 0..1 | Topology hiding information for a service API |  |

##### 8.3.4.2.6 Type: AccessControlPolicyListExt

Table 8.3.4.2.6-1: Definition of type AccessControlPolicyListExt

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description | Applicability |
| apiId | string | M | 1 | Identifier of the service API |  |
| NOTE: This data type also contains all the properties defined for AccessControlPolicyList data type. | | | | | |

##### 8.3.4.2.7 Type: TopologyHiding

Table 8.3.4.2.7-1: Definition of type TopologyHiding

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description | Applicability |
| apiId | string | M | 1 | Identifier of the service API |  |
| routingRules | array(RoutingRule) | M | 1..N | Routing rules |  |

#### 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 (NOTE) |  |
| API\_INVOKER\_AUTHORIZATION\_REVOKED | Events related to the revocation of the authorization of API invokers to access the service APIs. (NOTE) |  |
| API\_INVOKER\_UPDATED | Events related to API invoker profile updated to CAPIF. |  |
| API\_TOPOLOGY\_HIDING\_CREATED | Events related to the creation or update of the API topology hiding information of the service API after the service APIs are published |  |
| API\_TOPOLOGY\_HIDING\_REVOKED | Events related to the revocation of the API topology information of the service API after the service APIs are unpublished |  |
| NOTE: The present release does not specify further details (e.g. event filters) for this event. | | |

### 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. |
| 3 | Enhanced\_event\_report | This feature supports the enhanced event report including event reporting requirement and event reporting details as defined in subclause 5.4.2.2.2. |

## 8.4 CAPIF\_API\_Invoker\_Management\_API

### 8.4.1 API URI

The CAPIF\_API\_Invoker\_Management\_API service shall use the CAPIF\_API\_Invoker\_Management\_API.

The request URIs used in HTTP requests from the API invoker towards the CAPIF core function shall have the Resource URI 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 | /onboardedInvokers | POST | On-boards a new API invoker by creating an API invoker profile |
| Individual On-boarded API Invoker | /onboardedInvokers/{onboardingId} | DELETE | Off-boards an individual API invoker by deleting the associated API invoker profile identified by {onboardingId} |
| PUT | Updates the API invoker details of an individual API invoker identified by the {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/<apiVersion>/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 | Data Type | Definition |
| apiRoot | string | See clause 7.5 |
| apiVersion | string | See clause 8.4.1 |

##### 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 as part of the APIInvokerEnrolmentDetails which is provided in the response body, if requested in the POST request. |
| 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. |

Table 8.4.2.2.3.1-4: Headers supported by the 201 Response Code on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data type | P | Cardinality | Description |
| Location | string | M | 1 | Contains the URI of the newly created resource, according to the structure: {apiRoot}/api-invoker-management/<apiVersion>/onboardedInvokers/{onboardingId} |

###### 8.4.2.3.3.2 PUT

The PUT method allows updating the API invoker details of the onboarded API invoker. This method shall support the URI query parameters specified in table 8.4.2.3.3.2-1.

Table 8.4.2.3.3.2-1: 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 the table 8.4.2.3.3.2-2 and the response data structures and response codes specified in the table 8.4.2.3.3.2-3.

Table 8.4.2.3.3.2-2: Data structures supported by the PUT Request Body on this resource

|  |  |  |  |
| --- | --- | --- | --- |
| Data type | P | Cardinality | Description |
| APIInvokerEnrolmentDetails | M | 1 | Updated details of the API invoker and a notification destination URI for any update request related notifications. |

Table 8.4.2.3.3.2-3: Data structures supported by the PUT Response Body on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data type | P | Cardinality | Response  codes | Description |
| APIInvokerEnrolmentDetails | M | 1 | 200 OK | API invoker’s information updated successfully.  Updated details of the API invoker as part of the APIInvokerEnrolmentDetails, which is provided in the response body. |
| n/a |  |  | 202 Accepted | The CAPIF core has accepted the Update details request and is processing it. When processing is completed, the CAPIF core function will send a Notify\_Update\_Completion notification to the requesting API invoker. See sub clause 8.4.3.3. |

##### 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/<apiVersion>/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 | Data Type | Definition |
| apiRoot | string | See clause 7.5 |
| apiVersion | string | See clause 8.4.1 |
| onboardingId | string | 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 | Callback URI | HTTP method or custom operation | Description  (service operation) |
| Notify\_Onboarding\_Completion | {notificationDestination} | POST | Notify API invoker of on-boarding result |
| Notify\_Update\_Completion | {notificationDestination} | POST | Notify API invoker of update result details. |

#### 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.

Callback 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. |

#### 8.4.3.3 Notify\_Update\_Completion

##### 8.4.3.3.1 Description

Notify\_Update\_Completion is used by the CAPIF core function to notify of the update of API Invoker’s details result.

##### 8.4.3.3.2 Notification definition

The POST method shall be used for Notify\_Update\_Completion and the URI shall be the one provided by the API invoker during the API invoker details update request.

Callback URI: **{notificationDestination}**

This method shall support the URI query parameters specified in table 8.4.3.3.2-1.

Table 8.4.3.3.2-1: 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 8.4.3.3.2-2 and the response data structures and response codes specified in table 8.4.3.3.2-3.

Table 8.4.3.3.2-2: Data structures supported by the POST Request Body on this resource

|  |  |  |  |
| --- | --- | --- | --- |
| Data type | P | Cardinality | Description |
| OnboardingNotification | M | 1 | Notification with API Invoker’s details update result. |

Table 8.4.3.3.2-3: 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 |  |
| APIList | 8.4.4.2.4 | List of APIs |  |
| OnboardingInformation | 8.4.4.2.5 | On-boarding information of the API invoker |  |
| OnboardingNotification | 8.4.4.2.7 | Notification with on-boarding or update 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] |  |  |
| SupportedFeatures | 3GPP TS 29.571 [19] | Used to negotiate the applicability of optional features defined in table 8.4.6-1. |  |

#### 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. |  |
| 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 |
| 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 or API invoker update. 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 or API invoker update. |  |
| apiInvokerInformation | string | O | 0..1 | Generic information related to the API invoker such as details of the device or the application. |  |
| supportedFeatures | SupportedFeatures | O | 0..1 | Used to negotiate the supported optional features of the API as described in subclause 7.8.  This attribute shall be provided in the HTTP POST request and in the response of successful resource creation. |  |

##### 8.4.4.2.3 Type: Void

##### 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) | O | 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.  The subject field in the certificate shall be encoded with API invoker ID as Common Name as specified in IETF RFC 5280 [29]. |  |
| onboardingSecret | string | O | 0..1 | API invoker's onboarding secret, provided by the CAPIF core function. |  |

##### 8.4.4.2.6 Type: Void

##### 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 CAPIF\_Security\_API service shall use the CAPIF\_Security\_API.

The request URIs used in HTTP requests from the API invoker or the API exposing function towards the CAPIF core function shall have the Resource URI 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 | /trustedInvokers | n/a |  |
| Individual trusted API invoker | /trustedInvokers/{apiInvokerId} | GET | Retrieve authentication information of an API invoker |
| PUT | Create a security context for individual API invoker |
| DELETE | Revoke the authorization of the API invoker |
| /trustedInvokers/{apiInvokerId}/update | update (POST) | Update the security context (e.g. re-negotiate the security methods). |
| /trustedInvokers/{apiInvokerId}/delete | delete (POST) | Revoke the authorization of the API invoker for some APIs |
| /securities/{securityId}/token | token (POST) | Obtain the OAuth 2.0 authorization information |

#### 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/<apiVersion>/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 | Data Type | Definition |
| apiRoot | string | See clause 7.5 |
| apiVersion | string | See clause 8.5.1 |

##### 8.5.2.2.3 Resource Standard Methods

###### 8.5.2.2.3.1 Void

##### 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/<apiVersion>/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 | Data Type | Definition |
| apiRoot | string | See clause 7.5 |
| apiVersion | string | See clause 8.5.1 |
| apiInvokerId | string | Identifies 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.  (NOTE) |
| 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.  (NOTE) |
| NOTE: The query parameters "authenticationInfo" and "authorizationInfo" do not follow the related naming convention defined in subclause 7.5.1. These query parameters are however kept as currently defined in this specification for backward compatibility considerations. | | | | |

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 |

###### 8.5.2.3.3.3 PUT

This method shall support the URI query parameters specified in table 8.5.2.3.3.3-1.

Table 8.5.2.3.3.3-1: 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 8.5.2.3.3.3-2 and the response data structures and response codes specified in table 8.5.2.3.3.3-3.

Table 8.5.2.3.3.3-2: Data structures supported by the PUT 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 8.5.2.3.3.3-3: Data structures supported by the PUT 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. |

Table 8.5.2.3.3.3-4: Headers supported by the 201 Response Code on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data type | P | Cardinality | Description |
| Location | string | M | 1 | Contains the URI of the newly created resource, according to the structure: {apiRoot}/capif-security/v1/trustedInvokers/{apiInvokerId} |

##### 8.5.2.3.4 Resource Custom Operations

###### 8.5.2.3.4.1 Overview

Table 8.5.2.3.4.1-1: Custom operations

|  |  |  |  |
| --- | --- | --- | --- |
| Operation name | Custom operation URI | Mapped HTTP method | Description |
| update | /trustedInvokers/{apiInvokerId}/update | POST | Update the security instance (e.g. re-negotiate the security methods). |
| delete | /trustedInvokers/{apiInvokerId}/delete | POST | Revoke the authorization of the API invoker for some APIs |
| token | /securities/{securityId}/token | POST | Obtain the OAuth 2.0 authorization information |

###### 8.5.2.3.4.2 Operation: update

8.5.2.3.4.2.1 Description

This custom operation updates an existing Individual security instance resource in the CAPIF core function.

8.5.2.3.4.2.2 Operation Definition

This method shall support the URI query parameters specified in table 8.5.2.3.4.2.2-1.

Table 8.5.2.3.4.2.2-1: URI query parameters supported by the POST method on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data type | P | Cardinality | Description |
| n/a |  |  |  |  |

This operation shall support the request data structures specified in table 8.5.2.3.4.2.2-2 and the response data structure and response codes specified in table 8.5.2.3.4.2.2-3.

Table 8.5.2.3.4.2.2-2: 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 8.5.2.3.4.2.2-3: Data structures supported by the POST Response Body on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data type | P | Cardinality | Response codes | Description |
| ServiceSecurity | M | 1 | 200 OK | 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 |

###### 8.5.2.3.4.3 Operation: delete

8.5.2.3.4.3.1 Description

This custom operation revokes authorization for some service APIs of an existing Individual security instance resource in the CAPIF core function.

8.5.2.3.4.3.2 Operation Definition

This method shall support the URI query parameters specified in table 8.5.2.3.4.3.2-1.

Table 8.5.2.3.4.3.2-1: URI query parameters supported by the POST method on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data type | P | Cardinality | Description |
| n/a |  |  |  |  |

This operation shall support the request data structures specified in table 8.5.2.3.4.3.2-2 and the response data structure and response codes specified in table 8.5.2.3.4.3.2-3.

Table 8.5.2.3.4.3.2-2: Data structures supported by the POST Request Body on this resource

|  |  |  |  |
| --- | --- | --- | --- |
| Data type | P | Cardinality | Description |
| SecurityNotification | M | 1 | It includes a list of API identifiers for which authorization needs to be revoked for an API invoker. |

Table 8.5.2.3.4.3.2-3: Data structures supported by the POST Response Body on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data type | P | Cardinality | Response codes | Description |
| n/a |  |  | 204 No Content | Successful case.  The CAPIF core function revoked the authorization of the API invoker for the requested APIs. |

###### 8.5.2.3.4.4 Operation: token

8.5.2.3.4.4.1 Description

This custom operation obtains OAuth 2.0 authorization information from an existing Individual security instance resource in the CAPIF core function.

8.5.2.3.4.4.2 Operation Definition

This method shall support the URI query parameters specified in table 8.5.2.3.4.4.2-1.

Table 8.5.2.3.4.4.2-1: URI query parameters supported by the POST method on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data type | P | Cardinality | Description |
| n/a |  |  |  |  |

This operation shall support the request data structures specified in table 8.5.2.3.4.4.2-2 and the response data structure and response codes specified in table 8.5.2.3.4.4.2-3.

Table 8.5.2.3.4.4.2-2: Data structures supported by the POST Request Body on this resource

|  |  |  |  |
| --- | --- | --- | --- |
| Data type | P | Cardinality | Description |
| AccessTokenReq | M | 1 | This IE shall contain the request information for the access token request. |

Table 8.5.2.3.4.4.2-3: Data structures supported by the POST Response Body on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data type | P | Cardinality | Response codes | Description |
| AccessTokenRsp | M | 1 | 200 OK | This IE shall contain the access token response information. |
| AccessTokenErr | M | 1 | 400 Bad Request | See IETF RFC 6749 [23] subclause 5.2. |

### 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 | Callback 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.

Callback 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 |
| AccessTokenClaims | 8.5.4.2.6 | The claims data structure for the access token. |  |
| AccessTokenReq | 8.5.4.2.7 | Data type for carrying information related to access token request. |  |
| AccessTokenRsp | 8.5.4.2.8 | Data type for carrying information related to access token response. |  |
| SecurityInformation | 8.5.4.2.3 | Interface details and the security method |  |
| SecurityNotification | 8.5.4.2.5 | Revoked authorization notification details |  |
| 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. |  |

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 |
| DurationSec | 3GPP TS 29.122 [14] | Duration in seconds |  |
| SecurityMethod | Subclause 8.2.4.3.6 | Security method (e.g. PKI) |  |
| 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. |  |
| Uri | 3GPP TS 29.122 [14] |  |  |
| 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. |  |
| SupportedFeatures | 3GPP TS 29.571 [19] | Used to negotiate the applicability of optional features defined in table 8.5.6-1. |  |

#### 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 |
| securityInfo | array(SecurityInformation) | M | 1..N | Security information for each API interface. |  |
| 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 |
| supportedFeatures | SupportedFeatures | O | 0..1 | Used to negotiate the supported optional features of the API as described in subclause 7.8.  This attribute shall be provided in the HTTP POST request and in the response of successful resource creation. |  |

##### 8.5.4.2.3 Type: SecurityInformation

Table 8.5.4.2.3-1: Definition of type SecurityInformation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description | Applicability |
| interfaceDetails | InterfaceDescription | O | 0..1 | Details of the interface (NOTE) |  |
| aefId | string | O | 0..1 | AEF identifier (NOTE) |  |
| prefSecurityMethods | array(SecurityMethod) | M | 1..N | Security methods preferred by the API invoker for the API interface |  |
| selSecurityMethod | SecurityMethod | O | 0..1 | Supplied by the CAPIF core function, it indicates the selected security method for the API interface. If it is not provided, it means no common supported security method by the API invoker and the AEF, or the selected security method is not allowed by the local policy in the CAPIF core function. |  |
| authenticationInfo | string | O | 0..1 | Authentication related information |  |
| authorizationInfo | string | O | 0..1 | Authorization related information |  |
| NOTE:      Only one of the attributes "aefId" or "interfaceDetails" shall be included. | | | | | |

##### 8.5.4.2.4 Void

##### 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 |  |
| aefId | string | O | 0..1 | String identifying the AEF. |  |
| apiIds | array(string) | M | 1..N | Identifier of the service API |  |
| cause | Cause | M | 1 | The cause for revoking the API invoker authorization to the service API |  |

##### 8.5.4.2.6 Type: AccessTokenReq

Table 8.5.4.2.6-1: Definition of type AccessTokenReq

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| grant\_type | string | M | 1 | This IE shall contain the grant type as "client\_credentials" |
| client\_id | string | M | 1 | This IE shall contain the API invoker Identifier. |
| client\_secret | string | O | 0..1 | This IE when present shall contain the onboarding secret which is got during API invoker onboarding. |
| scope | string | O | 0..1 | This IE when present shall contain a list of AEF identifiers and its associated API names for which the access\_token is authorized for use.  It takes the format of 3gpp#aefId1:apiName1,apiName2,…apiNameX;aefId2:apiName1,apiName2,…apiNameY;…aefIdN:apiName1,apiName2,…apiNameZ  Using delimeter "#" after the discriminator "3gpp", ":" after AEF identifier, "," between API names and ";" between the last API name of the previous AEF identifier and the next AEF identifier. (NOTE 2)  Example: '3gpp#aef-jiangsu-nanjing:3gpp-monitoring-event,3gpp-as-session-with-qos;aef-zhejiang-hangzhou:3gpp-cp-parameter-provisioning,3gpp-pfd-management' |
| NOTE 1: This data structure shall not be treated as a JSON object. It shall be treated as a key, value pair data structure to be encoded using x-www-urlencoded format as specified in subclause 17.13.4.1 of W3C HTML 4.01 Specification [22].  NOTE 2: The scope may contain more space-delimited strings which further add additional access ranges to the scope, the definition of those additional strings is out of the scope of the present document. | | | | |

##### 8.5.4.2.7 Type: AccessTokenRsp

Table 8.5.4.2.7-1: Definition of type AccessTokenRsp

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| access\_token | string | M | 1 | This IE shall contain JWS Compact Serialized representation of the JWS signed JSON object containing AccessTokenClaims (see subclause 8.5.4.2.c). |
| token\_type | string | M | 1 | This IE shall contain the token type (i.e. "Bearer"). |
| expires\_in | DurationSec | M | 1 | This IE when present shall contain the number of seconds after which the access\_token is considered to be expired. |
| scope | string | O | 0..1 | This IE when present shall contain a list of AEF identifiers and its associated API names for which the access\_token is authorized for use.  It takes the format of 3gpp#aefId1:apiName1,apiName2,…apiNameX;aefId2:apiName1,apiName2,…apiNameY;…aefIdN:apiName1,apiName2,…apiNameZ  Using delimeter "#" after the discriminator "3gpp", ":" after AEF identifier, "," between API names and ";" between the last API name of the previous AEF identifier and the next AEF identifier. (NOTE)  Example: '3gpp#aef-jiangsu-nanjing:3gpp-monitoring-event,3gpp-as-session-with-qos;aef-zhejiang-hangzhou:3gpp-cp-parameter-provisioning,3gpp-pfd-management' |
| NOTE: The scope may contain more space-delimited strings which further add additional access ranges to the scope, the definition of those additional strings is out of the scope of the present document. | | | | |

##### 8.5.4.2.8 Type: AccessTokenClaims

Table 8.5.4.2.8-1: Definition of type AccessTokenClaims

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description |
| iss | string | M | 1 | This IE shall contain the API invoker Identifier. |
| scope | string | M | 1 | This IE shall contain a list of AEF identifiers and its associated API names for which the access\_token is authorized for use.  It takes the format of 3gpp#aefId1:apiName1,apiName2,…apiNameX;aefId2:apiName1,apiName2,…apiNameY;…aefIdN:apiName1,apiName2,…apiNameZ  Using delimeter "#" after the discriminator "3gpp", ":" after AEF identifier, "," between API names and ";" between the last API name of the previous AEF identifier and the next AEF identifier. (NOTE)  Example: '3gpp#aef-jiangsu-nanjing:3gpp-monitoring-event,3gpp-as-session-with-qos;aef-zhejiang-hangzhou:3gpp-cp-parameter-provisioning,3gpp-pfd-management' |
| exp | DurationSec | M | 1 | This IE shall contain the number of seconds after which the access\_token is considered to be expired. |
| NOTE: The scope may contain more space-delimited strings which further add additional access ranges to the scope, the definition of those additional strings is out of the scope of the present document. | | | | |

#### 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 CAPIF\_Access\_Control\_Policy\_API service shall use the CAPIF\_Access\_Control\_Policy\_API.

The request URIs used in HTTP requests from the API exposing function towards the CAPIF core function shall have the Resource URI 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 | /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/<apiVersion>/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 | Data Type | Definition |
| apiRoot | string | See clause 7.5 |
| apiVersion | string | See clause 8.6.1 |
| serviceApiId | string | Identifies 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 |
| aef-id | string | M | 1 | AEF identifier |
| api-invoker-id | string | O | 1 | String identifying the API invoker |
| supported-features | SupportedFeatures | O | 0..1 | To filter irrelevant responses related to unsupported features. |

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) | O | 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 | 0..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. |  |
| stopTime | 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 CAPIF\_Logging\_API\_Invocation\_API service shall use the CAPIF\_Logging\_API\_Invocation\_API.

The request URIs used in HTTP requests from the API exposing function towards the CAPIF core function shall have the Resource URI 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 | /{aefId}/logs | POST | Creates a new log entry for service API invocations |
| Individual log | /{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/<apiVersion>/{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 | Data Type | Definition |
| apiRoot | string | See clause 7.5 |
| apiVersion | string | See clause 8.7.1 |
| aefId | string | Identifies 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 |
| InvocationLog | M | 1 | 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 |
| InvocationLog | 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. |

Table 8.7.2.2.3.1-4: Headers supported by the 201 Response Code on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data type | P | Cardinality | Description |
| Location | string | M | 1 | Contains the URI of the newly created resource, according to the structure: {apiRoot}/api-invocation-logs/<apiVersion>/{aefId}/logs/{logId} |

##### 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 |
| InvocationLog | 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 |
| DateTime | 3GPP TS 29.122 [14] |  |  |
| InterfaceDescription | Subclause 8.2.4.2.3 | Details of the interface |  |
| SupportedFeatures | 3GPP TS 29.571 [19] | Used to negotiate the applicability of optional features defined in table 8.7.6-1. |  |
| Operation | Subclause 8.2.4.3.7 | HTTP operation |  |
| Protocol | Subclause 8.2.4.3.3 | API protocol |  |
| Uri | 3GPP TS 29.122 [14] |  |  |

#### 8.7.4.2 Structured data types

##### 8.7.4.2.1 Introduction

##### 8.7.4.2.2 Type: InvocationLog

Table 8.7.4.2.2-1: Definition of type InvocationLog

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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 |  |
| logs | array(Log) | M | 1..N | Service API invocation log |  |
| supportedFeatures | SupportedFeatures | O | 0..1 | Used to negotiate the supported optional features of the API as described in subclause 7.8.  This attribute shall be provided in the HTTP POST request and in the response of successful resource creation. |  |

##### 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. |  |
| apiName | string | M | 1 | Name of the API which was invoked, it is set as {apiName} part of the URI structure as defined in subclause 4.4 of 3GPP TS 29.501 [18]. |  |
| apiVersion | string | M | 1 | Version of the API which was invoked |  |
| resourceName | String | M | 1 | Name of the specific resource invoked |  |
| uri | Uri | O | 0..1 | Full URI of the API resource as defined in subclause 4.4 of 3GPP TS 29.501 [18]. |  |
| protocol | Protocol | M | 1 | Protocol invoked. |  |
| operation | Operation | C | 0..1 | Operation that was invoked on the API, only applicable for HTTP protocol. |  |
| result | string | M | 1 | For HTTP protocol, it contains HTTP status code of the invocation |  |
| invocationTime | DateTime | O | 0..1 | Date on which it was invoked |  |
| invocationLatency | DurationMs | O | 0..1 | Latency for the API invocation. |  |
| inputParameters | ANY TYPE  (NOTE) | O | 0..1 | List of input parameters |  |
| OutputParameters | ANY TYPE  (NOTE) | O | 0..1 | List of output parameters |  |
| srcInterface | InterfaceDescription | O | 0..1 | Interface description of the API invoker. |  |
| destInterface | InterfaceDescription | O | 0..1 | Interface description of the API invoked. |  |
| fwdInterface | string | O | 0..1 | It includes the node identifier (as defined in IETF RFC 7239 [20] of all forwarding entities between the API invoker and the AEF, concatenated with comma and space, e.g. 192.0.2.43:80, unknown:\_OBFport, 203.0.113.60 |  |
| NOTE: Any basic data type defined in OpenAPI 3.0.0 Specification [3] may be used. | | | | | |

#### 8.7.4.3 Simple data types and enumerations

##### 8.7.4.3.1 Introduction

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

##### 8.7.4.3.2 Simple data types

The simple data types defined in table 8.7.4.3.2-1 shall be supported.

Table 8.7.4.3.2-1: Simple data types

|  |  |  |  |
| --- | --- | --- | --- |
| **Type Name** | **Type Definition** | **Description** | **Applicability** |
| DurationMs | integer | Unsigned integer identifying a period of time in units of milliseconds. |  |

### 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 CAPIF\_Auditing\_API service shall use the CAPIF\_Auditing\_API.

The request URIs used in HTTP requests from the API management function towards the CAPIF core function shall have the Resource URI 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) | /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/<apiVersion>/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 | Data Type | Definition |
| apiRoot | string | See clause 7.5 |
| apiVersion | string | See clause 8.8.1 |

##### 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 |
| aef-id | string | O | 0..1 | String identifying the API exposing function |
| api-invoker-id | string | O | 0..1 | String identifying the API invoker which invoked the service API |
| time-range-start | DateTime | O | 0..1 | Start time of the invocation time range |
| time-range-end | DateTime | O | 0..1 | End time of the invocation time range |
| api-id | string | O | 0..1 | String identifying the API invoked. |
| api-name | string | O | 0..1 | API name, it is set as {apiName} part of the URI structure as defined in subclause 4.4 of 3GPP TS 29.501 [18]. |
| api-version | string | O | 0..1 | Version of the API which was invoked |
| protocol | Protocol | O | 0..1 | Protocol invoked |
| operation | Operation | O | 0..1 | Operation that was invoked on the API |
| result | string | O | 0..1 | HTTP status code of the invocation |
| resource-name | string | O | 0..1 | Name of the specific resource invoked |
| src-interface | InterfaceDescription | O | 0..1 | Interface description of the API invoker. |
| dest-interface | InterfaceDescription | O | 0..1 | Interface description of the API invoked. |
| supported-features | SupportedFeatures | O | 0..1 | To filter irrelevant responses related to unsupported features. |

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 |
| InvocationLog | O | 0..1 | 200 OK | Result of the query operation along with fetched service API invocation log data. |
| ProblemDetails | O | 0..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 |
| DateTime | 3GPP TS 29.122 [14] |  |  |
| InterfaceDescription | Subclause 8.2.4.2.3 | Interface description |  |
| InvocationLog | Subclause 8.7.4.2.2 | Logs of service API invocations stored on the CAPIF core function. |  |
| Operation | Subclause 8.2.4.3.7 | HTTP operation |  |
| ProblemDetails | 3GPP TS 29.122 [14] |  |  |
| Protocol | Subclause 8.2.4.3.3 | API protocol |  |

#### 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 |  |  |

## 8.9 CAPIF\_API\_Provider\_Management\_API

### 8.9.1 API URI

The CAPIF\_API\_Provider\_Management\_API service shall use the CAPIF\_API\_Provider\_Management\_API.

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

- The <apiName>shall be "api-provider-management".

- The <apiVersion> shall be "v1".

- The <apiSpecificSuffixes> shall be set as described in clause 8.9.2.

### 8.9.2 Resources

#### 8.9.2.1 Overview



Figure 8.9.2.1-1: Resource URI structure of the CAPIF\_API\_Provider\_Management\_API

Table 8.9.2.1-1 provides an overview of the resources and applicable HTTP methods.

Table 8.9.2.1-1: Resources and methods overview

|  |  |  |  |
| --- | --- | --- | --- |
| Resource name | Resource URI | HTTP method or custom operation | Description |
| All API Provider Domains Registrations | /registrations | POST | Registers a new API provider domain by creating an API provider domain with API provider domain functions profiles. |
| Individual API Provider Domain Registration | /registrations/{registrationId} | PUT | Updates an individual API provider domain identified by {registrationId} |
| DELETE | Deregisters an API provider domain by deleting the API provider domain and functions, identified by {registrationId}. |

#### 8.9.2.2 Resource: All API Provider Domains Registrations

##### 8.9.2.2.1 Description

The All API provider domains registrations resource represents all the API provider domains that are registered at a given CAPIF core function.

##### 8.9.2.2.2 Resource Definition

Resource URI: **{apiRoot}/api-provider-management/<apiVersion>/registrations**

This resource shall support the resource URI variables defined in table 8.9.2.2.2-1.

Table 8.9.2.2.2-1: Resource URI variables for this resource

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Definition |
| apiRoot | string | See clause 7.5 |
| apiVersion | string | See clause 8.9.1 |

##### 8.9.2.2.3 Resource Standard Methods

###### 8.9.2.2.3.1 POST

This method shall support the URI query parameters specified in table 8.9.2.2.3.1-1.

Table 8.9.2.2.3.1-1: 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 8.9.2.2.3.1-2 and the response data structures and response codes specified in table 8.9.2.2.3.1-3.

Table 8.9.2.2.3.1-2: Data structures supported by the POST Request Body on this resource

|  |  |  |  |
| --- | --- | --- | --- |
| Data type | P | Cardinality | Description |
| APIProviderEnrolmentDetails | M | 1 | Enrolment details of the API provider domain including individual API provider domain function details. |

Table 8.9.2.2.3.1-3: Data structures supported by the POST Response Body on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data type | P | Cardinality | Response  codes | Description |
| APIProviderEnrolmentDetails | M | 1 | 201 Created | API provider domain registered successfully  The URI of the created resource shall be returned in the "Location" HTTP header.  The list of successfully registered individual API provider domain functions, registration specific failure information of failed API provider domain function registrations, are included in APIProviderEnrolmentDetails which is provided in the response body. |

Table 8.9.2.2.3.1-4: Headers supported by the 201 Response Code on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data type | P | Cardinality | Description |
| Location | string | M | 1 | Contains the URI of the newly created resource, according to the structure: {apiRoot}/api-provider-management/<apiVersion>/registrations/{registrationId} |

##### 8.9.2.2.4 Resource Custom Operations

None.

#### 8.9.2.3 Resource: Individual API Provider Domain Registration

##### 8.9.2.3.1 Description

The Individual API Provide Domain Registration resource represents an individual API provider domain that is registered at a given CAPIF core function.

##### 8.9.2.3.2 Resource Definition

Resource URI: **{apiRoot}/api-provider-management/<apiVersion>/registrations/{registrationId}**

This resource shall support the resource URI variables defined in table 8.9.2.3.2-1.

Table 8.9.2.3.2-1: Resource URI variables for this resource

|  |  |  |
| --- | --- | --- |
| Name | Data Type | Definition |
| apiRoot | string | See clause 7.5 |
| apiVersion | string | See clause 8.9.1 |
| registrationId | string | Identifies an individual registered API Provider domain resource |

##### 8.9.2.3.3 Resource Standard Methods

###### 8.9.2.3.3.1 PUT

The PUT method allows updating the registered API provider domain’s detail. The properties "apiProviderDomainId”, and “supportedFeatures” shall remain unchanged from previously provided values. This method shall support the URI query parameters specified in table 8.9.2.3.3.1-1.

Table 8.9.2.3.3.1-1: 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 the table 8.9.2.3.3.1-2 and the response data structures and response codes specified in the table 8.9.2.3.3.1-3.

Table 8.9.2.3.3.1-2: Data structures supported by the PUT Request Body on this resource

|  |  |  |  |
| --- | --- | --- | --- |
| Data type | P | Cardinality | Description |
| APIProviderEnrolmentDetails | M | 1 | Updated details of the API provider domain. |

Table 8.9.2.3.3.1-3: Data structures supported by the PUT Response Body on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data type | P | Cardinality | Response  codes | Description |
| APIProviderEnrolmentDetails | M | 1 | 200 OK | API provider domain’s information updated successfully.  Updated details of the API provider domain is part of the APIProviderEnrolmentDetails, which is provided in the response body. The list of successfully updated individual API provider domain functions, registration update specific failure information of failed API provider domain function registration updates, are included in APIProviderEnrolmentDetails which is provided in the response body. |

###### 8.9.2.3.3.2 DELETE

This method shall support the URI query parameters specified in table 8.9.2.3.3.2-1.

Table 8.9.2.3.3.2-1: URI query parameters supported by the DELETE method on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data type | P | Cardinality | Description |
| n/a |  |  |  |  |

This method shall support the response codes specified in table 8.9.2.3.3.2-2 and the response data structures and response codes specified in table 8.9.2.3.3.2-3.

Table 8.9.2.3.3.2-2: Data structures supported by the DELETE Request Body on this resource

|  |  |  |  |
| --- | --- | --- | --- |
| Data type | P | Cardinality | Description |
| n/a |  |  |  |

Table 8.9.2.3.3.2-3: Data structures supported by the DELETE Response Body on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data type | P | Cardinality | Response  codes | Description |
| n/a |  |  | 204 No Content | The individual registered API provider domain matching the registrationId is deleted. All the individual API provider domain functions of the API provider domain are deleted. |

##### 8.9.2.3.4 Resource Custom Operations

None.

### 8.9.3 Notifications

None.

### 8.9.4 Data Model

#### 8.9.4.1 General

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

Table 8.9.4.1-1 specifies the data types defined specifically for the CAPIF\_API\_Provider\_Management\_API service.

Table 8.9.4.1-1: CAPIF\_API\_Provider\_Management\_API specific Data Types

|  |  |  |  |
| --- | --- | --- | --- |
| Data type | Section defined | Description | Applicability |
| APIProviderEnrolmentDetails | 8.9.4.2.2 | API provider domain’s enrolment details. |  |
| APIProviderFunctionDetails | 8.9.4.2.3 | API Provider domain function’s details. |  |
| RegistrationInformation | 8.9.4.2.4 | Registration information of the individual API provider domain function. |  |

Table 8.9.4.1-2 specifies data types re-used by the CAPIF\_API\_Provider\_Management\_API service.

Table 8.9.4.1-2: Re-used Data Types

|  |  |  |  |
| --- | --- | --- | --- |
| Data type | Reference | Comments | Applicability |
| SupportedFeatures | 3GPP TS 29.571 [19] | Used to negotiate the applicability of optional features defined in table 8.4.6-1. |  |

#### 8.9.4.2 Structured data types

##### 8.9.4.2.1 Introduction

##### 8.9.4.2.2 Type: APIProviderEnrolmentDetails

Table 8.9.4.2.2-1: Definition of type APIProviderEnrolmentDetails

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description | Applicability |
| apiProvDomId | string | O | 0..1 | API provider domain ID assigned by the CAPIF core function to the API management function while registering the API provider domain. Shall not be present in the HTTP POST request from the API management function to the CAPIF core function, to on-board itself. Shall be present in all other HTTP requests and responses. |  |
| regSec | string | M | 1 | Security information necessary for the CAPIF core function to validate the registration of the API provider domain. Shall be present in HTTP POST request from API management function to CAPIF core function for API provider domain registration. |  |
| apiProvFuncs | array(APIProviderFunctionDetails) | O | 1..N | A list of individual API provider domain functions details. When included by the API management function in the HTTP request message, it lists the API provider domain functions that the API management function intends to register/update in registration or update registration procedure. When included by the CAPIF core function in the HTTP response message, it lists the API domain functions details that are registered or updated successfully. |  |
| apiProvDomInfo | string | O | 0..1 | Generic information related to the API provider domain such as details of the API provider applications. |  |
| suppFeat | SupportedFeatures | O | 0..1 | Used to negotiate the supported optional features of the API as described in subclause 7.8.  This attribute shall be provided in the HTTP POST request and in the response of successful resource creation. |  |
| failReason | string | O | 0..1 | Registration or update specific failure information of failed API provider domain function registrations.  Shall be present in the HTTP response body if atleast one of the API provider domain function registration or update registration fails. |  |

##### 8.9.4.2.3 Type: APIProviderFunctionDetails

Table 8.9.4.2.3-1: Definition of type APIProviderFunctionDetails

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description | Applicability |
| apiProvFuncId | string | O | 0..1 | API provider domain functionID assigned by the CAPIF core function to the API provider domain function while registering/updating the API provider domain. Shall not be present in the HTTP POST request from the API management function to the CAPIF core function, to register itself. Shall be present in all other HTTP requests and responses. |  |
| regInfo | RegistrationInformation | M | 1 | Information necessary for the CAPIF core function to register the API provider domain function.  This information shall be present in HTTP POST/PUT request from API management function to CAPIF core function for API provider domain registration. In the HTTP response message from CAPIF core function, shall include the updated registration information for API provider domain function. |  |
| apiProvFuncRole | ApiProviderFuncRole | M | 1 | Role of API provider domain function.  The role shall be present in the HTTP POST/PUT request that the API management function intends to register/update the API provider domain function as. CAPIF core function shall register the role of API provider domain function as per the request. |  |
| apiProvFuncInfo | string | O | 0..1 | Generic information related to the API provider domain function such as details of the API provider applications. |  |

##### 8.9.4.2.4 Type: RegistrationInformation

Table 8.9.4.2.4-1: Definition of type RegistrationInformation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description | Applicability |
| apiProvPubKey | string | M | 1 | Public Key of API Provider domain function. |  |
| apiProvCert | string | O | 0..1 | API provider domain function’s generic client certificate |  |

#### 8.9.4.3 Simple data types and enumerations

##### 8.9.4.3.1 Introduction

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

##### 8.9.4.3.2 Simple data types

The simple data types defined in table 8.9.4.3.2-1 shall be supported.

Table 8.9.4.3.2-1: Simple data types

|  |  |  |  |
| --- | --- | --- | --- |
| Type Name | Type Definition | Description | Applicability |
| n/a |  |  |  |

##### 8.9.4.3.3 Enumeration: ApiProviderFuncRole

Table 8.9.4.3.3-1: Enumeration ApiProviderFuncRole

|  |  |  |
| --- | --- | --- |
| Enumeration value | Description | Applicability |
| AEF | API provider function is API Exposing Function. |  |
| APF | API provider function is API Publishing Function. |  |
| AMF | API provider function is API Management Function. |  |

### 8.9.5 Error Handling

General error responses are defined in clause 7.7 of this specification.

### 8.9.6 Feature negotiation

General feature negotiation procedures are defined in clause 7.8 of this specification.

## 8.10 CAPIF\_Routing\_Info\_API

### 8.10.1 API URI

The CAPIF\_Routing\_Info\_API service shall use the CAPIF\_Routing\_Info\_API.

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

- The <apiName>shall be "capif-routing-info".

- The <apiVersion> shall be "v1".

- The <apiSpecificSuffixes> shall be set as described in subclause 8.10.2.

### 8.10.2 Resources

#### 8.10.2.1 Overview



Figure 8.10.2.1-1: Resource URI structure of the CAPIF\_Routing\_Info\_API

Table 8.10.2.1-1 provides an overview of the resources and applicable HTTP methods.

Table 8.10.2.1-1: Resources and methods overview

|  |  |  |  |
| --- | --- | --- | --- |
| Resource name | Resource URI | HTTP method or custom operation | Description |
| Individual Service API routing info | /service-apis/{serviceApiId} | GET | Retrieves the API routing information for a published service API and API exposing function which applies the topology hiding. |

#### 8.10.2.2 Resource: Individual Service API routing info

##### 8.10.2.2.1 Description

The API Routing Information resource represents the API routing information for the service API per API Exposing Function.

##### 8.10.2.2.2 Resource Definition

Resource URI: **{apiRoot}/capif-routing-info/<apiVersion>/service-apis/{serviceApiId}**

This resource shall support the resource URI variables defined in table 8.10.2.2.2-1.

Table 8.10.2.2.2-1: Resource URI variables for this resource

|  |  |  |
| --- | --- | --- |
| Name | Data type | Definition |
| apiRoot | string | See subclause 7.5 |
| apiVersion | string | See subclause 8.10.1 |
| serviceApiId | string | Identifies an individual published service API |

##### 8.10.2.2.3 Resource Standard Methods

###### 8.10.2.2.3.1 GET

This method shall support the URI query parameters specified in table 8.10.2.2.3.1-1.

Table 8.10.2.2.3.1-1: URI query parameters supported by the GET method on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data type | P | Cardinality | Description |
| aef-id | string | M | 1 | AEF identifier |
| supp-feat | SupportedFeatures | O | 0..1 | To filter irrelevant responses related to unsupported features. |

This method shall support the request data structures specified in table 8.10.2.2.3.1-2 and the response data structures and response codes specified in table 8.10.2.2.3.1-3.

Table 8.10.2.2.3.1-2: Data structures supported by the GET Request Body on this resource

|  |  |  |  |
| --- | --- | --- | --- |
| Data type | P | Cardinality | Description |
| n/a |  |  |  |

Table 8.10.2.2.3.1-3: Data structures supported by the GET Response Body on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data type | P | Cardinality | Response  Codes | Description |
| RoutingInfo | M | 1 | 200 OK | The Routing information applicable for the service API requested. |

##### 8.10.2.2.4 Resource Custom Operations

None.

### 8.10.3 Notifications

None.

### 8.10.4 Data Model

#### 8.10.4.1 General

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

Table 8.10.4.1-1 specifies the data types defined specifically for the CAPIF\_Routing\_Info\_API service.

Table 8.10.4.1-1: CAPIF\_Routing\_Info\_API specific Data Types

|  |  |  |  |
| --- | --- | --- | --- |
| Data type | Section defined | Description | Applicability |
| RoutingInfo | 8.10.4.2.2 | API routing information |  |
| RoutingRule | 8.10.4.2.3 | API routing rule |  |
| Ipv6AddressRange | 8.10.4.2.4 | IP address range |  |

Table 8.10.4.1-2 specifies data types re-used by the CAPIF\_Routing\_Info\_API service.

Table 8.10.4.1-2: Re-used Data Types

|  |  |  |  |
| --- | --- | --- | --- |
| Data type | Reference | Comments | Applicability |
| Ipv4AddressRange | 3GPP TS 29.510 [28] |  |  |
| Ipv6Addr | 3GPP TS 29.122 [14] |  |  |
| SupportedFeatures | 3GPP TS 29.571 [19] |  |  |

#### 8.10.4.2 Structured data types

##### 8.10.4.2.1 Introduction

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

##### 8.10.4.2.2 Type: RoutingInfo

Table 8.10.4.2.2-1: Definition of type RoutingInfo

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description | Applicability |
| routingRules | array(RoutingRule) | M | 1..N | Routing rules |  |

##### 8.10.4.2.3 Type: RoutingRule

Table 8.10.4.2.3-1: Definition of type RoutingRule

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description | Applicability |
| ipv4AddrRanges | array(Ipv4AddressRange) | O | 1..N | The IPv4 address range for the API invocation source IP address. (NOTE) |  |
| ipv6AddrRanges | array(Ipv6AddressRange) | O | 1..N | The IPv6 address range for the API invocation source IP address. (NOTE) |  |
| aefProfile | AefProfile | M | 1 | The target AEF profile |  |
| NOTE: If no IP address range is provided, it means the service API invocation from any source IP address can be routed to the target AEF. | | | | | |

##### 8.10.4.2.4 Type: Ipv6AddressRange

Table 8.10.4.2.4-1: Definition of type Ipv6AddressRange

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description | Applicability |
| start | Ipv6Addr | M | 1 | First value identifying the start of an IPv6 address range |  |
| end | Ipv6Addr | M | 1 | Last value identifying the end of an IPv6 address range |  |

#### 8.10.4.3 Simple data types and enumerations

None.

### 8.10.5 Error Handling

General error responses are defined in subclause 7.7.

### 8.10.6 Feature negotiation

General feature negotiation procedures are defined in subclause 7.8.

Table 8.10.6-1: Supported Features

|  |  |  |
| --- | --- | --- |
| **Feature number** | **Feature Name** | **Description** |
| n/a |  |  |

# 9 AEF API Definition

## 9.1 AEF\_Security\_API

### 9.1.1 API URI

The AEF\_Security\_API service shall use the AEF\_Security\_API.

The request URIs used in HTTP requests from the API invoker towards the API exposing function shall have the Resource URI structure as defined in subclause 7.5 with the following clarifications:

- The <apiName>shall be "aef-security".

- The <apiVersion> shall be "v1".

- The <custOpName> shall be set as described in subclause 9.1.2a.

### 9.1.2 Resources

There is no resource defined for this API.

##### 9.1.2.2.4 Resource Custom Operations

None.

### 9.1.2a Custom Operations without associated resources

#### 9.1.2a.1 Overview

Custom operations used for this API are summarized in table 9.1.2a.1-1. "{apiRoot}" and “<apiVersion>” are set as described in clause 7.5 and clause 9.1.1 respectively.

Table 9.1.2a.1-1: Custom operations without associated resources

|  |  |  |  |
| --- | --- | --- | --- |
| Operation name | Custom operation URI | Mapped HTTP method | Description |
| check-authentication | /check-authentication | POST | Check authentication request. |
| revoke-authentication | /revoke-authorization | POST | Revoke authorization for service APIs. |

#### 9.1.2a.2 Operation: check-authentication

##### 9.1.2a.2.1 Description

This custom operation allows the API invoker to confirm from the API exposing function, that necessary authentication data is available to authenticate the API invoker on API invocation.

##### 9.1.2a.2.2 Operation Definition

This method shall support the URI query parameters specified in table 9.1.2a.2.2-1.

Table 9.1.2a.2.2-1: URI query parameters supported by the POST method on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data type | P | Cardinality | Description |
| n/a |  |  |  |  |

This operation shall support the request and response data structures, and response codes specified in tables 9.1.2a.2.2-2 and 9.1.2a.2.2-3.

Table 9.1.2a.2.2-2: Data structures supported by the POST Request Body on this operation

|  |  |  |  |
| --- | --- | --- | --- |
| Data type | P | Cardinality | Description |
| CheckAuthenticationReq | M | 1 | Authentication check request data |

Table 9.1.2a.2.2-3: Data structures supported by the POST Response Body on this operation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data type | P | Cardinality | Response  codes | Description |
| CheckAuthenticationRsp | M | 1 | 200 OK | The request was successful. |

#### 9.1.2a.3 Operation: revoke-authorization

##### 9.1.2a.3.1 Description

This custom operation allows the CAPIF core function to request the API exposing function to revoke the authorization of the API invoker for the indicated service APIs.

##### 9.1.2a.3.2 Operation Definition

This method shall support the URI query parameters specified in table 9.1.2a.3.2-1.

Table 9.1.2a.3.2-1: URI query parameters supported by the POST method on this resource

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Data type | P | Cardinality | Description |
| n/a |  |  |  |  |

This operation shall support the request and response data structures, and response codes specified in tables 9.1.2a.2.3-2 and 9.1.2a.2.3-3.

Table 9.1.2a.2.3-2: Data structures supported by the POST Request Body on this operation

|  |  |  |  |
| --- | --- | --- | --- |
| Data type | P | Cardinality | Description |
| RevokeAuthorizationReq | M | 1 | Authorization revocation request data |

Table 9.1.2a.2.3-3: Data structures supported by the POST Response Body on this operation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data type | P | Cardinality | Response  codes | Description |
| RevokeAuthorizationRsp | M | 1 | 200 OK | The request was successful. |

### 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\_Security\_API service.

Table 9.1.4.1-1: AEF\_Security\_API specific Data Types

|  |  |  |  |
| --- | --- | --- | --- |
| Data type | Section defined | Description | Applicability |
| CheckAuthenticationReq | Subclause 9.1.4.2.2 | Authentication check request data |  |
| CheckAuthenticationRsp | Subclause 9.1.4.2.3 | Authentication check response data |  |
| RevokeAuthorizationReq | Subclause 9.1.4.2.4 | Authorization revocation request data |  |
| RevokeAuthorizationRsp | Subclause 9.1.4.2.5 | Authorization revocation response data |  |

Table 9.1.4.1-2 specifies data types re-used by the AEF\_Security\_API service.

Table 9.1.4.1-2: Re-used Data Types

|  |  |  |  |
| --- | --- | --- | --- |
| Data type | Reference | Comments | Applicability |
| SecurityNotification | Subclause 8.5.4.2.5 | Information about the revoked APIs |  |
| SupportedFeatures | 3GPP TS 29.571 [19] | Used to negotiate the applicability of optional features defined in table 9.1.6-1. |  |

#### 9.1.4.2 Structured data types

##### 9.1.4.2.1 Introduction

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

##### 9.1.4.2.2 Type: CheckAuthenticationReq

Table 9.1.4.2.2-1: Definition of type CheckAuthenticationReq

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description | Applicability |
| apiInvokerId | string | M | 1 | API invoker ID assigned by the CAPIF core function to the API invoker while on-boarding the API invoker. |  |
| supportedFeatures | SupportedFeatures | M | 1 | Used to negotiate the supported optional features of the API as described in subclause 7.8. |  |

##### 9.1.4.2.3 Type: CheckAuthenticationRsp

Table 9.1.4.2.3-1: Definition of type CheckAuthenticationRsp

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description | Applicability |
| supportedFeatures | SupportedFeatures | M | 1 | Used to negotiate the supported optional features of the API as described in subclause 7.8. |  |

##### 9.1.4.2.4 Type: RevokeAuthorizationReq

Table 9.1.4.2.4-1: Definition of type RevokeAuthorizationReq

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description | Applicability |
| revokeInfo | SecurityNotification | M | 1 | It contains detailed revocation information. |  |
| supportedFeatures | SupportedFeatures | M | 1 | Used to negotiate the supported optional features of the API as described in subclause 7.8. |  |

##### 9.1.4.2.5 Type: RevokeAuthorizationRsp

Table 9.1.4.2.5-1: Definition of type RevokeAuthorizationRsp

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Attribute name | Data type | P | Cardinality | Description | Applicability |
| supportedFeatures | SupportedFeatures | M | 1 | Used to negotiate the supported optional features of the API as described in subclause 7.8. |  |

#### 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].

## 10.2 CAPIF-1/1e security

Secure communication between API invoker and CAPIF core function over CAPIF-1/1e reference points, 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].

## 10.3 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 Initiate\_Authentication service operation of the AEF\_Security\_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 Annex is based on the OpenAPI 3.0.0 specification [3] and provides corresponding representations of all APIs defined in the present specification, in YAML format.

This Annex shall take precedence when being discrepant to other parts of the specification with respect to the encoding of information elements and methods within the API.

NOTE: The semantics and procedures, as well as conditions, e.g. for the applicability and allowed combinations of attributes or values, not expressed in the OpenAPI definitions but defined in other parts of the specification also apply.

Informative copies of the OpenAPI specification file contained in this 3GPP Technical Specification are available on a Git-based repository that uses the GitLab software version control system (see clause 5B of the 3GPP TR 21.900 [27] and clause 5.3.1 of the 3GPP TS 29.501 [18] for further information).

# A.2 CAPIF\_Discover\_Service\_API

openapi: 3.0.0

info:

title: CAPIF\_Discover\_Service\_API

description: |

API for discovering service APIs.

© 2020, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).

All rights reserved.

version: "1.1.0"

externalDocs:

description: 3GPP TS 29.222 V16.3.0 Common API Framework for 3GPP Northbound APIs

url: http://www.3gpp.org/ftp/Specs/archive/29\_series/29.222/

servers:

- url: '{apiRoot}/service-apis/v1'

variables:

apiRoot:

default: https://example.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: api-invoker-id

in: query

description: String identifying the API invoker assigned by the CAPIF core function. It also represents the CCF identifier in the CAPIF-6/6e interface.

required: true

schema:

type: string

- name: api-name

in: query

description: API name, it is set as {apiName} part of the URI structure as defined in subclause 4.4 of 3GPP TS 29.501 [18].

schema:

type: string

- name: api-version

in: query

description: API major version the URI (e.g. v1).

schema:

type: string

- name: comm-type

in: query

description: Communication type used by the API (e.g. REQUEST\_RESPONSE).

schema:

$ref: 'TS29222\_CAPIF\_Publish\_Service\_API.yaml#/components/schemas/CommunicationType'

- name: protocol

in: query

description: Protocol used by the API.

schema:

$ref: 'TS29222\_CAPIF\_Publish\_Service\_API.yaml#/components/schemas/Protocol'

- name: aef-id

in: query

description: AEF identifer.

schema:

type: string

- name: data-format

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'

- name: api-cat

in: query

description: The service API category to which the service API belongs to.

schema:

type: string

- name: supported-features

in: query

description: Features supported by the NF consumer for the CAPIF Discover Service API.

schema:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/SupportedFeatures'

- name: api-supported-features

in: query

description: Features supported by the discovered service API indicated by api-name parameter. This may only be present if api-name query parameter is present.

schema:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/SupportedFeatures'

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'

'404':

$ref: 'TS29122\_CommonData.yaml#/components/responses/404'

'406':

$ref: 'TS29122\_CommonData.yaml#/components/responses/406'

'414':

$ref: 'TS29122\_CommonData.yaml#/components/responses/414'

'429':

$ref: 'TS29122\_CommonData.yaml#/components/responses/429'

'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: object

properties:

serviceAPIDescriptions:

type: array

items:

$ref: 'TS29222\_CAPIF\_Publish\_Service\_API.yaml#/components/schemas/ServiceAPIDescription'

minItems: 1

description: Description of the service API as published by the service. Each service API description shall include AEF profiles matching the filter criteria.

# A.3 CAPIF\_Publish\_Service\_API

openapi: 3.0.0

info:

title: CAPIF\_Publish\_Service\_API

description: |

API for publishing service APIs.

© 2022, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).

All rights reserved.

version: "1.1.2"

externalDocs:

description: 3GPP TS 29.222 V16.9.0 Common API Framework for 3GPP Northbound APIs

url: http://www.3gpp.org/ftp/Specs/archive/29\_series/29.222/

servers:

- url: '{apiRoot}/published-apis/v1'

variables:

apiRoot:

default: https://example.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:

type: string

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'

headers:

Location:

description: 'Contains the URI of the newly created resource, according to the structure: {apiRoot}/published-apis/v1/{apfId}/service-apis/{serviceApiId}'

required: true

schema:

type: string

'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'

'411':

$ref: 'TS29122\_CommonData.yaml#/components/responses/411'

'413':

$ref: 'TS29122\_CommonData.yaml#/components/responses/413'

'415':

$ref: 'TS29122\_CommonData.yaml#/components/responses/415'

'429':

$ref: 'TS29122\_CommonData.yaml#/components/responses/429'

'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:

type: string

responses:

'200':

description: Definition of all service API(s) published by the API publishing function.

content:

application/json:

schema:

type: array

items:

$ref: '#/components/schemas/ServiceAPIDescription'

minItems: 0

'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'

'406':

$ref: 'TS29122\_CommonData.yaml#/components/responses/406'

'429':

$ref: 'TS29122\_CommonData.yaml#/components/responses/429'

'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:

type: string

- name: apfId

in: path

required: true

schema:

type: string

responses:

'200':

description: Definition of individual 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'

'403':

$ref: 'TS29122\_CommonData.yaml#/components/responses/403'

'404':

$ref: 'TS29122\_CommonData.yaml#/components/responses/404'

'406':

$ref: 'TS29122\_CommonData.yaml#/components/responses/406'

'429':

$ref: 'TS29122\_CommonData.yaml#/components/responses/429'

'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:

type: string

- name: apfId

in: path

required: true

schema:

type: string

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'

'411':

$ref: 'TS29122\_CommonData.yaml#/components/responses/411'

'413':

$ref: 'TS29122\_CommonData.yaml#/components/responses/413'

'415':

$ref: 'TS29122\_CommonData.yaml#/components/responses/415'

'429':

$ref: 'TS29122\_CommonData.yaml#/components/responses/429'

'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:

type: string

- name: apfId

in: path

required: true

schema:

type: string

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'

'429':

$ref: 'TS29122\_CommonData.yaml#/components/responses/429'

'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 Type for representations

ServiceAPIDescription:

type: object

properties:

apiName:

type: string

description: API name, it is set as {apiName} part of the URI structure as defined in subclause 4.4 of 3GPP TS 29.501.

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 and in the HTTP GET response from the CAPIF core function to the API invoker (discovery API).

aefProfiles:

type: array

items:

$ref: '#/components/schemas/AefProfile'

minItems: 1

description: AEF profile information, which includes the exposed API details (e.g. protocol).

description:

type: string

description: Text description of the API

supportedFeatures:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/SupportedFeatures'

shareableInfo:

$ref: '#/components/schemas/ShareableInformation'

serviceAPICategory:

type: string

apiSuppFeats:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/SupportedFeatures'

pubApiPath:

$ref: '#/components/schemas/PublishedApiPath'

ccfId:

type: string

description: CAPIF core function identifier.

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/SecurityMethod'

minItems: 1

description: Security methods supported by the interface, it take precedence over the security methods provided in AefProfile, for this specific interface.

oneOf:

- required: [ipv4Addr]

- required: [ipv6Addr]

AefProfile:

type: object

properties:

aefId:

type: string

description: Identifier of the API exposing function

versions:

type: array

items:

$ref: '#/components/schemas/Version'

minItems: 1

description: API version

protocol:

$ref: '#/components/schemas/Protocol'

dataFormat:

$ref: '#/components/schemas/DataFormat'

securityMethods:

type: array

items:

$ref: '#/components/schemas/SecurityMethod'

minItems: 1

description: Security methods supported by the AEF

domainName:

type: string

description: Domain to which API belongs to

interfaceDescriptions:

type: array

items:

$ref: '#/components/schemas/InterfaceDescription'

minItems: 1

description: Interface details

required:

- aefId

- versions

oneOf:

- required: [domainName]

- required: [interfaceDescriptions]

Resource:

type: object

properties:

resourceName:

type: string

description: Resource name

commType:

$ref: '#/components/schemas/CommunicationType'

uri:

type: string

description: Relative URI of the API resource, it is set as {apiSpecificResourceUriPart} part of the URI structure as defined in subclause 4.4 of 3GPP TS 29.501.

custOpName:

type: string

description: it is set as {custOpName} part of the URI structure for a custom operation associated with a resource as defined in subclause 4.4 of 3GPP TS 29.501.

operations:

type: array

items:

$ref: '#/components/schemas/Operation'

minItems: 1

description: Supported HTTP methods for the API resource. Only applicable when the protocol in AefProfile indicates HTTP.

description:

type: string

description: Text description of the API resource

required:

- resourceName

- commType

- uri

CustomOperation:

type: object

properties:

commType:

$ref: '#/components/schemas/CommunicationType'

custOpName:

type: string

description: it is set as {custOpName} part of the URI structure for a custom operation without resource association as defined in subclause 4.4 of 3GPP TS 29.501.

operations:

type: array

items:

$ref: '#/components/schemas/Operation'

minItems: 1

description: Supported HTTP methods for the API resource. Only applicable when the protocol in AefProfile indicates HTTP.

description:

type: string

description: Text description of the custom operation

required:

- commType

- custOpName

Version:

type: object

properties:

apiVersion:

type: string

description: API major version in URI (e.g. v1)

expiry:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/DateTime'

resources:

type: array

items:

$ref: '#/components/schemas/Resource'

minItems: 1

description: Resources supported by the API.

custOperations:

type: array

items:

$ref: '#/components/schemas/CustomOperation'

minItems: 1

description: Custom operations without resource association.

required:

- apiVersion

ShareableInformation:

type: object

properties:

isShareable:

type: boolean

description: Set to "true" indicates that the service API and/or the service API category can be shared to the list of CAPIF provider domain information. Otherwise set to "false".

capifProvDoms:

type: array

items:

type: string

minItems: 1

description: List of CAPIF provider domains to which the service API information to be shared.

required:

- isShareable

PublishedApiPath:

type: object

properties:

ccfIds:

type: array

items:

type: string

minItems: 1

description: A list of CCF identifiers where the service API is already published.

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

SecurityMethod:

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

Operation:

anyOf:

- type: string

enum:

- GET

- POST

- PUT

- PATCH

- DELETE

- 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

- GET: HTTP GET method

- POST: HTTP POST method

- PUT: HTTP PUT method

- PATCH: HTTP PATCH method

- DELETE: HTTP DELETE method

# A.4 CAPIF\_Events\_API

openapi: 3.0.0

info:

title: CAPIF\_Events\_API

description: |

API for event subscription management.

© 2020, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).

All rights reserved.

version: "1.1.0"

externalDocs:

description: 3GPP TS 29.222 V16.3.0 Common API Framework for 3GPP Northbound APIs

url: http://www.3gpp.org/ftp/Specs/archive/29\_series/29.222/

servers:

- url: '{apiRoot}/capif-events/v1'

variables:

apiRoot:

default: https://example.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'

'404':

$ref: 'TS29122\_CommonData.yaml#/components/responses/404'

'411':

$ref: 'TS29122\_CommonData.yaml#/components/responses/411'

'413':

$ref: 'TS29122\_CommonData.yaml#/components/responses/413'

'415':

$ref: 'TS29122\_CommonData.yaml#/components/responses/415'

'429':

$ref: 'TS29122\_CommonData.yaml#/components/responses/429'

'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'

headers:

Location:

description: 'Contains the URI of the newly created resource, according to the structure: {apiRoot}/capif-events/v1/{subscriberId}/subscriptions/{subscriptionId}'

required: true

schema:

type: string

'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'

'411':

$ref: 'TS29122\_CommonData.yaml#/components/responses/411'

'413':

$ref: 'TS29122\_CommonData.yaml#/components/responses/413'

'415':

$ref: 'TS29122\_CommonData.yaml#/components/responses/415'

'429':

$ref: 'TS29122\_CommonData.yaml#/components/responses/429'

'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'

'429':

$ref: 'TS29122\_CommonData.yaml#/components/responses/429'

'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

eventFilters:

type: array

items:

$ref: '#/components/schemas/CAPIFEventFilter'

minItems: 1

description: Subscribed event filters

eventReq:

$ref: 'TS29523\_Npcf\_EventExposure.yaml#/components/schemas/ReportingInformation'

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'

supportedFeatures:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/SupportedFeatures'

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'

eventDetail:

$ref: '#/components/schemas/CAPIFEventDetail'

required:

- subscriptionId

- events

CAPIFEventFilter:

type: object

properties:

apiIds:

type: array

items:

type: string

minItems: 1

description: Identifier of the service API

apiInvokerIds:

type: array

items:

type: string

minItems: 1

description: Identity of the API invoker

aefIds:

type: array

items:

type: string

minItems: 1

description: Identifier of the API exposing function

CAPIFEventDetail:

type: object

properties:

serviceAPIDescriptions:

type: array

items:

$ref: 'TS29222\_CAPIF\_Publish\_Service\_API.yaml#/components/schemas/ServiceAPIDescription'

minItems: 1

description: Description of the service API as published by the APF.

apiIds:

type: array

items:

type: string

minItems: 1

description: Identifier of the service API

apiInvokerIds:

type: array

items:

type: string

minItems: 1

description: Identity of the API invoker

accCtrlPolList:

$ref: '#/components/schemas/AccessControlPolicyListExt'

invocationLogs:

type: array

items:

$ref: 'TS29222\_CAPIF\_Logging\_API\_Invocation\_API.yaml#/components/schemas/InvocationLog'

minItems: 1

description: Invocation logs.

apiTopoHide:

$ref: '#/components/schemas/TopologyHiding'

AccessControlPolicyListExt:

allOf:

- $ref: 'TS29222\_CAPIF\_Access\_Control\_Policy\_API.yaml#/components/schemas/AccessControlPolicyList'

- type: object

properties:

apiId:

type: string

required:

- apiId

TopologyHiding:

type: object

properties:

apiId:

type: string

routingRules:

type: array

items:

$ref: 'TS29222\_CAPIF\_Routing\_Info\_API.yaml#/components/schemas/RoutingRule'

minItems: 1

required:

- apiId

- routingRules

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

- API\_INVOKER\_UPDATED

- API\_TOPOLOGY\_HIDING\_CREATED

- API\_TOPOLOGY\_HIDING\_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.

- API\_INVOKER\_UPDATED: Events related to API invoker profile updated to CAPIF.

- API\_TOPOLOGY\_HIDING\_CREATED: Events related to the creation or update of the API topology hiding information of the service APIs after the service APIs are published.

- API\_TOPOLOGY\_HIDING\_REVOKED: Events related to the revocation of the API topology hiding information of the service APIs after the service APIs are unpublished.

# A.5 CAPIF\_API\_Invoker\_Management\_API

openapi: 3.0.0

info:

title: CAPIF\_API\_Invoker\_Management\_API

description: |

API for API invoker management.

© 2022, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).

All rights reserved.

version: "1.1.1"

externalDocs:

description: 3GPP TS 29.222 V16.10.0 Common API Framework for 3GPP Northbound APIs

url: http://www.3gpp.org/ftp/Specs/archive/29\_series/29.222/

servers:

- url: '{apiRoot}/api-invoker-management/v1'

variables:

apiRoot:

default: https://example.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'

'404':

$ref: 'TS29122\_CommonData.yaml#/components/responses/404'

'411':

$ref: 'TS29122\_CommonData.yaml#/components/responses/411'

'413':

$ref: 'TS29122\_CommonData.yaml#/components/responses/413'

'415':

$ref: 'TS29122\_CommonData.yaml#/components/responses/415'

'429':

$ref: 'TS29122\_CommonData.yaml#/components/responses/429'

'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'

headers:

Location:

description: 'Contains the URI of the newly created resource, according to the structure: {apiRoot}/api-invoker-management/v1/onboardedInvokers/{onboardingId}'

required: true

schema:

type: string

'202':

description: The CAPIF core has accepted the Onboarding request and is processing it.

'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'

'411':

$ref: 'TS29122\_CommonData.yaml#/components/responses/411'

'413':

$ref: 'TS29122\_CommonData.yaml#/components/responses/413'

'415':

$ref: 'TS29122\_CommonData.yaml#/components/responses/415'

'429':

$ref: 'TS29122\_CommonData.yaml#/components/responses/429'

'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'

'429':

$ref: 'TS29122\_CommonData.yaml#/components/responses/429'

'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: Updates an individual API invoker details.

parameters:

- name: onboardingId

in: path

description: String identifying an individual on-boarded API invoker resource

required: true

schema:

type: string

requestBody:

description: representation of the API invoker details to be updated in CAPIF core function

required: true

content:

application/json:

schema:

$ref: '#/components/schemas/APIInvokerEnrolmentDetails'

callbacks:

notificationDestination:

'{request.body#/notificationDestination}':

post:

description: Notify the API Invoker about the API invoker update completion

requestBody: # contents of the callback message

required: true

content:

application/json:

schema:

$ref: '#/components/schemas/OnboardingNotification'

responses:

'204':

description: No Content (successful API invoker update 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'

'404':

$ref: 'TS29122\_CommonData.yaml#/components/responses/404'

'411':

$ref: 'TS29122\_CommonData.yaml#/components/responses/411'

'413':

$ref: 'TS29122\_CommonData.yaml#/components/responses/413'

'415':

$ref: 'TS29122\_CommonData.yaml#/components/responses/415'

'429':

$ref: 'TS29122\_CommonData.yaml#/components/responses/429'

'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:

'200':

description: API invoker details updated successfully

content:

application/json:

schema:

$ref: '#/components/schemas/APIInvokerEnrolmentDetails'

'202':

description: The CAPIF core has accepted the API invoker update details request and is processing it.

'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'

'411':

$ref: 'TS29122\_CommonData.yaml#/components/responses/411'

'413':

$ref: 'TS29122\_CommonData.yaml#/components/responses/413'

'415':

$ref: 'TS29122\_CommonData.yaml#/components/responses/415'

'429':

$ref: 'TS29122\_CommonData.yaml#/components/responses/429'

'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, provided by the CAPIF core function.

onboardingSecret:

type: string

description: The API Invoker’s onboarding secret, provided by the CAPIF core function.

required:

- apiInvokerPublicKey

APIList:

type: object

description: Represents a list of APIs.

properties:

serviceAPIDescriptions:

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.

readOnly: true

onboardingInformation:

$ref: '#/components/schemas/OnboardingInformation'

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:

$ref: '#/components/schemas/APIList'

apiInvokerInformation:

type: string

description: Generic information related to the API invoker such as details of the device or the application.

supportedFeatures:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/SupportedFeatures'

required:

- onboardingInformation

- notificationDestination

description: Information about the API Invoker that requested to onboard

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

description: |

API for CAPIF security management.

© 2020, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).

All rights reserved.

version: "1.0.3"

externalDocs:

description: 3GPP TS 29.222 V15.7.0 Common API Framework for 3GPP Northbound APIs

url: http://www.3gpp.org/ftp/Specs/archive/29\_series/29.222/

servers:

- url: '{apiRoot}/capif-security/v1'

variables:

apiRoot:

default: https://example.com

description: apiRoot as defined in subclause 7.5 of 3GPP TS 29.222.

paths:

/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'

'406':

$ref: 'TS29122\_CommonData.yaml#/components/responses/406'

'414':

$ref: 'TS29122\_CommonData.yaml#/components/responses/414'

'429':

$ref: 'TS29122\_CommonData.yaml#/components/responses/429'

'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:

parameters:

- name: apiInvokerId

in: path

description: Identifier of an individual API invoker

required: true

schema:

type: string

requestBody:

description: create a security context for an API invoker

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'

'404':

$ref: 'TS29122\_CommonData.yaml#/components/responses/404'

'411':

$ref: 'TS29122\_CommonData.yaml#/components/responses/411'

'413':

$ref: 'TS29122\_CommonData.yaml#/components/responses/413'

'415':

$ref: 'TS29122\_CommonData.yaml#/components/responses/415'

'429':

$ref: 'TS29122\_CommonData.yaml#/components/responses/429'

'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'

headers:

Location:

description: 'Contains the URI of the newly created resource, according to the structure: {apiRoot}/capif-security/v1/trustedInvokers/{apiInvokerId}'

required: true

schema:

type: string

'400':

$ref: 'TS29122\_CommonData.yaml#/components/responses/400'

'401':

$ref: 'TS29122\_CommonData.yaml#/components/responses/401'

'403':

$ref: 'TS29122\_CommonData.yaml#/components/responses/403'

'411':

$ref: 'TS29122\_CommonData.yaml#/components/responses/411'

'413':

$ref: 'TS29122\_CommonData.yaml#/components/responses/413'

'414':

$ref: 'TS29122\_CommonData.yaml#/components/responses/414'

'415':

$ref: 'TS29122\_CommonData.yaml#/components/responses/415'

'429':

$ref: 'TS29122\_CommonData.yaml#/components/responses/429'

'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'

'429':

$ref: 'TS29122\_CommonData.yaml#/components/responses/429'

'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}/update:

post:

parameters:

- name: apiInvokerId

in: path

description: Identifier of an individual API invoker

required: true

schema:

type: string

requestBody:

description: Update the security context (e.g. re-negotiate the security methods).

required: true

content:

application/json:

schema:

$ref: '#/components/schemas/ServiceSecurity'

responses:

'200':

description: Successful updated.

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'

'411':

$ref: 'TS29122\_CommonData.yaml#/components/responses/411'

'413':

$ref: 'TS29122\_CommonData.yaml#/components/responses/413'

'415':

$ref: 'TS29122\_CommonData.yaml#/components/responses/415'

'429':

$ref: 'TS29122\_CommonData.yaml#/components/responses/429'

'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}/delete:

post:

parameters:

- name: apiInvokerId

in: path

description: Identifier of an individual API invoker

required: true

schema:

type: string

requestBody:

description: Revoke the authorization of the API invoker for APIs.

required: true

content:

application/json:

schema:

$ref: '#/components/schemas/SecurityNotification'

responses:

'204':

description: Successful revoked.

'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'

'411':

$ref: 'TS29122\_CommonData.yaml#/components/responses/411'

'413':

$ref: 'TS29122\_CommonData.yaml#/components/responses/413'

'415':

$ref: 'TS29122\_CommonData.yaml#/components/responses/415'

'429':

$ref: 'TS29122\_CommonData.yaml#/components/responses/429'

'500':

$ref: 'TS29122\_CommonData.yaml#/components/responses/500'

'503':

$ref: 'TS29122\_CommonData.yaml#/components/responses/503'

default:

$ref: 'TS29122\_CommonData.yaml#/components/responses/default'

/securities/{securityId}/token:

post:

parameters:

- name: securityId

in: path

description: Identifier of an individual API invoker

required: true

schema:

type: string

requestBody:

required: true

content:

application/x-www-form-urlencoded:

schema:

$ref: '#/components/schemas/AccessTokenReq'

responses:

'200':

description: Successful Access Token Request

content:

application/json:

schema:

$ref: '#/components/schemas/AccessTokenRsp'

'400':

description: Error in the Access Token Request

content:

application/json:

schema:

$ref: '#/components/schemas/AccessTokenErr'

components:

schemas:

ServiceSecurity:

type: object

properties:

securityInfo:

type: array

items:

$ref: '#/components/schemas/SecurityInformation'

minimum: 1

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'

supportedFeatures:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/SupportedFeatures'

required:

- securityInfo

- notificationDestination

SecurityInformation:

type: object

properties:

interfaceDetails:

$ref: 'TS29222\_CAPIF\_Publish\_Service\_API.yaml#/components/schemas/InterfaceDescription'

aefId:

type: string

description: Identifier of the API exposing function

prefSecurityMethods:

type: array

items:

$ref: 'TS29222\_CAPIF\_Publish\_Service\_API.yaml#/components/schemas/SecurityMethod'

minItems: 1

description: Security methods preferred by the API invoker for the API interface.

selSecurityMethod:

$ref: 'TS29222\_CAPIF\_Publish\_Service\_API.yaml#/components/schemas/SecurityMethod'

authenticationInfo:

type: string

description: Authentication related information

authorizationInfo:

type: string

description: Authorization related information

required:

- prefSecurityMethods

oneOf:

- required: [interfaceDetails]

- required: [aefId]

SecurityNotification:

type: object

properties:

apiInvokerId:

type: string

description: String identifying the API invoker assigned by the CAPIF core function

aefId:

type: string

description: String identifying the AEF.

apiIds:

type: array

items:

type: string

minItems: 1

description: Identifier of the service API

cause:

$ref: '#/components/schemas/Cause'

required:

- apiInvokerId

- apiIds

- cause

AccessTokenReq:

format: x-www-form-urlencoded

properties:

grant\_type:

type: string

enum:

- client\_credentials

client\_id:

type: string

client\_secret:

type: string

scope:

type: string

required:

- grant\_type

- client\_id

AccessTokenRsp:

type: object

properties:

access\_token:

type: string

description: JWS Compact Serialized representation of JWS signed JSON object (AccessTokenClaims)

token\_type:

type: string

enum:

- Bearer

expires\_in:

$ref: 'TS29122\_CommonData.yaml#/components/schemas/DurationSec'

scope:

type: string

required:

- access\_token

- token\_type

- expires\_in

AccessTokenClaims:

type: object

properties:

iss:

type: string

scope:

type: string

exp:

$ref: 'TS29122\_CommonData.yaml#/components/schemas/DurationSec'

required:

- iss

- scope

- exp

AccessTokenErr:

type: object

properties:

error:

type: string

enum:

- invalid\_request

- invalid\_client

- invalid\_grant

- unauthorized\_client

- unsupported\_grant\_type

- invalid\_scope

error\_description:

type: string

error\_uri:

type: string

required:

- error

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

description: |

API for access control policy.

© 2020, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).

All rights reserved.

version: "1.0.3"

externalDocs:

description: 3GPP TS 29.222 V15.6.0 Common API Framework for 3GPP Northbound APIs

url: http://www.3gpp.org/ftp/Specs/archive/29\_series/29.222/

servers:

- url: '{apiRoot}/access-control-policy/v1'

variables:

apiRoot:

default: https://example.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: aef-id

in: query

required: true

description: Identifier of the AEF

schema:

type: string

- name: api-invoker-id

in: query

description: Identifier of the API invoker

schema:

type: string

- name: supported-features

in: query

description: To filter irrelevant responses related to unsupported features

schema:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/SupportedFeatures'

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'

'406':

$ref: 'TS29122\_CommonData.yaml#/components/responses/406'

'414':

$ref: 'TS29122\_CommonData.yaml#/components/responses/414'

'429':

$ref: 'TS29122\_CommonData.yaml#/components/responses/429'

'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

description: |

API for invocation logs.

© 2020, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).

All rights reserved.

version: "1.0.2"

externalDocs:

description: 3GPP TS 29.222 V15.6.0 Common API Framework for 3GPP Northbound APIs

url: http://www.3gpp.org/ftp/Specs/archive/29\_series/29.222/

servers:

- url: '{apiRoot}/api-invocation-logs/v1'

variables:

apiRoot:

default: https://example.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/InvocationLog'

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/InvocationLog'

headers:

Location:

description: 'Contains the URI of the newly created resource, according to the structure: {apiRoot}/api-invocation-logs/v1/{aefId}/logs/{logId}'

required: true

schema:

type: string

'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'

'411':

$ref: 'TS29122\_CommonData.yaml#/components/responses/411'

'413':

$ref: 'TS29122\_CommonData.yaml#/components/responses/413'

'415':

$ref: 'TS29122\_CommonData.yaml#/components/responses/415'

'429':

$ref: 'TS29122\_CommonData.yaml#/components/responses/429'

'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:

InvocationLog:

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

logs:

type: array

items:

$ref: '#/components/schemas/Log'

minItems: 1

description: Service API invocation log

supportedFeatures:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/SupportedFeatures'

required:

- aefId

- apiInvokerId

- logs

Log:

type: object

properties:

apiId:

type: string

description: String identifying the API invoked.

apiName:

type: string

description: Name of the API which was invoked, it is set as {apiName} part of the URI structure as defined in subclause 4.4 of 3GPP TS 29.501.

apiVersion:

type: string

description: Version of the API which was invoked

resourceName:

type: string

description: Name of the specific resource invoked

uri:

$ref: 'TS29122\_CommonData.yaml#/components/schemas/Uri'

protocol:

$ref: 'TS29222\_CAPIF\_Publish\_Service\_API.yaml#/components/schemas/Protocol'

operation:

$ref: 'TS29222\_CAPIF\_Publish\_Service\_API.yaml#/components/schemas/Operation'

result:

type: string

description: For HTTP protocol, it contains HTTP status code of the invocation

invocationTime:

$ref: 'TS29122\_CommonData.yaml#/components/schemas/DateTime'

invocationLatency:

$ref: '#/components/schemas/DurationMs'

inputParameters:

description: List of input parameters. Can be any value - string, number, boolean, array or object.

outputParameters:

description: List of output parameters. Can be any value - string, number, boolean, array or object.

srcInterface:

$ref: 'TS29222\_CAPIF\_Publish\_Service\_API.yaml#/components/schemas/InterfaceDescription'

destInterface:

$ref: 'TS29222\_CAPIF\_Publish\_Service\_API.yaml#/components/schemas/InterfaceDescription'

fwdInterface:

type: string

description: It includes the node identifier (as defined in IETF RFC 7239 of all forwarding entities between the API invoker and the AEF, concatenated with comma and space, e.g. 192.0.2.43:80, unknown:\_OBFport, 203.0.113.60

required:

- apiId

- apiName

- apiVersion

- resourceName

- protocol

- result

DurationMs:

type: integer

description: Unsigned integer identifying a period of time in units of milliseconds.

minimum: 0

# A.9 CAPIF\_Auditing\_API

openapi: 3.0.0

info:

title: CAPIF\_Auditing\_API

description: |

API for auditing.

© 2019, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).

All rights reserved.

version: "1.0.1"

externalDocs:

description: 3GPP TS 29.222 V15.4.0 Common API Framework for 3GPP Northbound APIs

url: http://www.3gpp.org/ftp/Specs/archive/29\_series/29.222/

servers:

- url: '{apiRoot}/logs/v1'

variables:

apiRoot:

default: https://example.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: aef-id

in: query

description: String identifying the API exposing function.

schema:

type: string

- name: api-invoker-id

in: query

description: String identifying the API invoker which invoked the service API.

schema:

type: string

- name: time-range-start

in: query

description: Start time of the invocation time range.

schema:

$ref: 'TS29122\_CommonData.yaml#/components/schemas/DateTime'

- name: time-range-end

in: query

description: End time of the invocation time range.

schema:

$ref: 'TS29122\_CommonData.yaml#/components/schemas/DateTime'

- name: api-id

in: query

description: String identifying the API invoked.

schema:

type: string

- name: api-name

in: query

description: API name, it is set as {apiName} part of the URI structure as defined in subclause 4.4 of 3GPP TS 29.501.

schema:

type: string

- name: api-version

in: query

description: Version of the API which was invoked.

schema:

type: string

- name: protocol

in: query

description: Protocol invoked.

schema:

$ref: 'TS29222\_CAPIF\_Publish\_Service\_API.yaml#/components/schemas/Protocol'

- name: operation

in: query

description: Operation that was invoked on the API.

schema:

$ref: 'TS29222\_CAPIF\_Publish\_Service\_API.yaml#/components/schemas/Operation'

- name: result

in: query

description: Result or output of the invocation.

schema:

type: string

- name: resource-name

in: query

description: Name of the specific resource invoked.

schema:

type: string

- name: src-interface

in: query

description: Interface description of the API invoker.

content:

application/json:

schema:

$ref: 'TS29222\_CAPIF\_Publish\_Service\_API.yaml#/components/schemas/InterfaceDescription'

- name: dest-interface

in: query

description: Interface description of the API invoked.

content:

application/json:

schema:

$ref: 'TS29222\_CAPIF\_Publish\_Service\_API.yaml#/components/schemas/InterfaceDescription'

- name: supported-features

in: query

description: To filter irrelevant responses related to unsupported features

schema:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/SupportedFeatures'

responses:

'200':

description: Result of the query operation along with fetched service API invocation log data.

content:

application/json:

schema:

$ref: 'TS29222\_CAPIF\_Logging\_API\_Invocation\_API.yaml#/components/schemas/InvocationLog'

'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'

'406':

$ref: 'TS29122\_CommonData.yaml#/components/responses/406'

'414':

$ref: 'TS29122\_CommonData.yaml#/components/responses/414'

'429':

$ref: 'TS29122\_CommonData.yaml#/components/responses/429'

'500':

$ref: 'TS29122\_CommonData.yaml#/components/responses/500'

'503':

$ref: 'TS29122\_CommonData.yaml#/components/responses/503'

default:

$ref: 'TS29122\_CommonData.yaml#/components/responses/default'

# A.10 AEF\_Security\_API

openapi: 3.0.0

info:

title: AEF\_Security\_API

description: |

API for AEF security management.

© 2019, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).

All rights reserved.

version: "1.0.1"

externalDocs:

description: 3GPP TS 29.222 V15.4.0 Common API Framework for 3GPP Northbound APIs

url: http://www.3gpp.org/ftp/Specs/archive/29\_series/29.222/

servers:

- url: '{apiRoot}/aef-security/v1'

variables:

apiRoot:

default: https://example.com

description: apiRoot as defined in subclause 7.5 of 3GPP TS 29.222.

paths:

/check-authentication:

post:

summary: Check authentication.

requestBody:

required: true

content:

application/json:

schema:

$ref: '#/components/schemas/CheckAuthenticationReq'

responses:

'200':

description: The request was successful.

content:

application/json:

schema:

$ref: '#/components/schemas/CheckAuthenticationRsp'

'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'

'411':

$ref: 'TS29122\_CommonData.yaml#/components/responses/411'

'413':

$ref: 'TS29122\_CommonData.yaml#/components/responses/413'

'415':

$ref: 'TS29122\_CommonData.yaml#/components/responses/415'

'429':

$ref: 'TS29122\_CommonData.yaml#/components/responses/429'

'500':

$ref: 'TS29122\_CommonData.yaml#/components/responses/500'

'503':

$ref: 'TS29122\_CommonData.yaml#/components/responses/503'

default:

$ref: 'TS29122\_CommonData.yaml#/components/responses/default'

/revoke-authorization:

post:

summary: Revoke authorization.

requestBody:

required: true

content:

application/json:

schema:

$ref: '#/components/schemas/RevokeAuthorizationReq'

responses:

'200':

description: The request was successful.

content:

application/json:

schema:

$ref: '#/components/schemas/RevokeAuthorizationRsp'

'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'

'411':

$ref: 'TS29122\_CommonData.yaml#/components/responses/411'

'413':

$ref: 'TS29122\_CommonData.yaml#/components/responses/413'

'415':

$ref: 'TS29122\_CommonData.yaml#/components/responses/415'

'429':

$ref: 'TS29122\_CommonData.yaml#/components/responses/429'

'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:

CheckAuthenticationReq:

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.

supportedFeatures:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/SupportedFeatures'

required:

- apiInvokerId

- supportedFeatures

CheckAuthenticationRsp:

type: object

properties:

supportedFeatures:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/SupportedFeatures'

required:

- supportedFeatures

RevokeAuthorizationReq:

type: object

properties:

revokeInfo:

$ref: 'TS29222\_CAPIF\_Security\_API.yaml#/components/schemas/SecurityNotification'

supportedFeatures:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/SupportedFeatures'

required:

- revokeInfo

- supportedFeatures

RevokeAuthorizationRsp:

type: object

properties:

supportedFeatures:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/SupportedFeatures'

required:

- supportedFeatures

# A.11 CAPIF\_API\_Provider\_Management\_API

openapi: 3.0.0

info:

title: CAPIF\_API\_Provider\_Management\_API

description: |

API for API provider domain functions management.

© 2020, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).

All rights reserved.

version: "1.0.0"

externalDocs:

description: 3GPP TS 29.222 V16.3.0 Common API Framework for 3GPP Northbound APIs

url: http://www.3gpp.org/ftp/Specs/archive/29\_series/29.222/

servers:

- url: '{apiRoot}/api-provider-management/v1'

variables:

apiRoot:

default: https://example.com

description: apiRoot as defined in subclause 7.5 of 3GPP TS 29.222

paths:

/registrations:

post:

description: Registers a new API Provider domain with API provider domain functions profiles.

requestBody:

required: true

content:

application/json:

schema:

$ref: '#/components/schemas/APIProviderEnrolmentDetails'

responses:

'201':

description: API provider domain registered successfully

content:

application/json:

schema:

$ref: '#/components/schemas/APIProviderEnrolmentDetails'

headers:

Location:

description: 'Contains the URI of the newly created resource, according to the structure: {apiRoot}/api-provider-management/v1/registrations/{registrationId}'

required: true

schema:

type: string

'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'

'411':

$ref: 'TS29122\_CommonData.yaml#/components/responses/411'

'413':

$ref: 'TS29122\_CommonData.yaml#/components/responses/413'

'415':

$ref: 'TS29122\_CommonData.yaml#/components/responses/415'

'429':

$ref: 'TS29122\_CommonData.yaml#/components/responses/429'

'500':

$ref: 'TS29122\_CommonData.yaml#/components/responses/500'

'503':

$ref: 'TS29122\_CommonData.yaml#/components/responses/503'

default:

$ref: 'TS29122\_CommonData.yaml#/components/responses/default'

/registrations/{registrationId}:

delete:

description: Deregisters API provider domain by deleting API provider domain and functions.

parameters:

- name: registrationId

in: path

description: String identifying an registered API provider domain resource

required: true

schema:

type: string

responses:

'204':

description: The API provider domain matching registrationId 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'

'429':

$ref: 'TS29122\_CommonData.yaml#/components/responses/429'

'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: Updates an API provider domain's registration details.

parameters:

- name: registrationId

in: path

description: String identifying an registered API provider domain resource

required: true

schema:

type: string

requestBody:

description: representation of the API provider domain registration details to be updated in CAPIF core function

required: true

content:

application/json:

schema:

$ref: '#/components/schemas/APIProviderEnrolmentDetails'

responses:

'200':

description: API provider domain registration details updated successfully

content:

application/json:

schema:

$ref: '#/components/schemas/APIProviderEnrolmentDetails'

'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'

'411':

$ref: 'TS29122\_CommonData.yaml#/components/responses/411'

'413':

$ref: 'TS29122\_CommonData.yaml#/components/responses/413'

'415':

$ref: 'TS29122\_CommonData.yaml#/components/responses/415'

'429':

$ref: 'TS29122\_CommonData.yaml#/components/responses/429'

'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:

APIProviderEnrolmentDetails:

type: object

properties:

apiProvDomId:

type: string

description: API provider domain ID assigned by the CAPIF core function to the API management function while registering the API provider domain. Shall not be present in the HTTP POST request from the API Management function to the CAPIF core function, to on-board itself. Shall be present in all other HTTP requests and responses.

readOnly: true

regSec:

type: string

description: Security information necessary for the CAPIF core function to validate the registration of the API provider domain. Shall be present in HTTP POST request from API management function to CAPIF core function for API provider domain registration.

apiProvFuncs:

type: array

items:

$ref: '#/components/schemas/APIProviderFunctionDetails'

minItems: 1

description: A list of individual API provider domain functions details. When included by the API management function in the HTTP request message, it lists the API provider domain functions that the API management function intends to register/update in registration or update registration procedure. When included by the CAPIF core function in the HTTP response message, it lists the API domain functions details that are registered or updated successfully.

apiProvDomInfo:

type: string

description: Generic information related to the API provider domain such as details of the API provider applications.

suppFeat:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/SupportedFeatures'

failReason:

type: string

description: Registration or update specific failure information of failed API provider domain function registrations.Shall be present in the HTTP response body if atleast one of the API provider domain function registration or update registration fails.

required:

- regSec

APIProviderFunctionDetails:

type: object

properties:

apiProvFuncId:

type: string

description: API provider domain functionID assigned by the CAPIF core function to the API provider domain function while registering/updating the API provider domain. Shall not be present in the HTTP POST request from the API management function to the CAPIF core function, to register itself. Shall be present in all other HTTP requests and responses.

regInfo:

$ref: '#/components/schemas/RegistrationInformation'

apiProvFuncRole:

$ref: '#/components/schemas/ApiProviderFuncRole'

apiProvFuncInfo:

type: string

description: Generic information related to the API provider domain function such as details of the API provider applications.

required:

- regInfo

- apiProvFuncRole

RegistrationInformation:

type: object

properties:

apiProvPubKey:

type: string

description: Public Key of API Provider domain function.

apiProvCert:

type: string

description: API provider domain function's client certificate

required:

- apiProvPubKey

ApiProviderFuncRole:

anyOf:

- type: string

enum:

- AEF

- APF

- AMF

- type: string

description: >

This string provides forward-compatiblity 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

- AEF: API provider function is API Exposing Function.

- APF: API provider function is API Publishing Function.

- AMF: API Provider function is API Management Function.

# A.12 CAPIF\_Routing\_Info\_API

openapi: 3.0.0

info:

title: CAPIF\_Routing\_Info\_API

description: |

API for Routing information.

© 2020, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).

All rights reserved.

version: "1.0.0"

externalDocs:

description: 3GPP TS 29.222 V16.3.0 Common API Framework for 3GPP Northbound APIs

url: http://www.3gpp.org/ftp/Specs/archive/29\_series/29.222/

servers:

- url: '{apiRoot}/capif-routing-info/v1'

variables:

apiRoot:

default: https://example.com

description: apiRoot as defined in subclause 7.5 of 3GPP TS 29.222

paths:

/service-apis/{serviceApiId}:

get:

description: Retrieves the API routing information.

parameters:

- name: serviceApiId

in: path

description: Identifier of a published service API

required: true

schema:

type: string

- name: aef-id

in: query

required: true

description: Identifier of the AEF

schema:

type: string

- name: supp-feat

in: query

required: false

description: To filter irrelevant responses related to unsupported features

schema:

$ref: 'TS29571\_CommonData.yaml#/components/schemas/SupportedFeatures'

responses:

'200':

description: OK.

content:

application/json:

schema:

$ref: '#/components/schemas/RoutingInfo'

'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'

'406':

$ref: 'TS29122\_CommonData.yaml#/components/responses/406'

'414':

$ref: 'TS29122\_CommonData.yaml#/components/responses/414'

'429':

$ref: 'TS29122\_CommonData.yaml#/components/responses/429'

'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:

RoutingInfo:

type: object

properties:

routingRules:

type: array

items:

$ref: '#/components/schemas/RoutingRule'

minItems: 1

required:

- routingRules

RoutingRule:

type: object

properties:

ipv4AddrRanges:

type: array

items:

$ref: 'TS29510\_Nnrf\_NFManagement.yaml#/components/schemas/Ipv4AddressRange'

minItems: 1

ipv6AddrRanges:

type: array

items:

$ref: '#/components/schemas/Ipv6AddressRange'

minItems: 1

aefProfile:

$ref: 'TS29222\_CAPIF\_Publish\_Service\_API.yaml#/components/schemas/AefProfile'

required:

- aefProfile

Ipv6AddressRange:

type: object

properties:

start:

$ref: 'TS29122\_CommonData.yaml#/components/schemas/Ipv6Addr'

end:

$ref: 'TS29122\_CommonData.yaml#/components/schemas/Ipv6Addr'

required:

- start

- end

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 |
| 2018-12 | CT#82 | CP-183109 | 0047 |  | F | Correct server definition | 15.2.0 |
| 2018-12 | CT#82 | CP-183109 | 0027 | 2 | F | Security adaptation for Nnef northbound APIs with CAPIF | 15.2.0 |
| 2018-12 | CT#82 | CP-183109 | 0045 | 1 | F | Correct security API name in subclause 5.6.2.1 | 15.2.0 |
| 2018-12 | CT#82 | CP-183109 | 0046 | 1 | F | Remove Event operations from CAPIF\_Publish\_API | 15.2.0 |
| 2018-12 | CT#82 | CP-183109 | 0048 |  | F | Correct CAPIF services | 15.2.0 |
| 2018-12 | CT#82 | CP-183109 | 0049 | 2 | F | Correct api name and service name for CAPIF\_Publish\_Service\_API | 15.2.0 |
| 2018-12 | CT#82 | CP-183109 | 0050 | 2 | F | Correct api name and service name for CAPIF\_Discover\_Service\_API | 15.2.0 |
| 2018-12 | CT#82 | CP-183109 | 0051 | 4 | F | Correct CAPIF\_Publish\_Service\_API | 15.2.0 |
| 2018-12 | CT#82 | CP-183109 | 0052 | 1 | F | Correct CAPIF\_Discover\_Service\_API | 15.2.0 |
| 2018-12 | CT#82 | CP-183109 | 0053 | 4 | F | Correct CAPIF\_Logging\_API\_Invocation\_API | 15.2.0 |
| 2018-12 | CT#82 | CP-183109 | 0054 | 3 | F | Correct CAPIF\_Auditing\_API | 15.2.0 |
| 2018-12 | CT#82 | CP-183109 | 0055 | 2 | F | Correct CAPIF\_Security\_API | 15.2.0 |
| 2018-12 | CT#82 | CP-183109 | 0055 | 3 | F | Correct CAPIF\_Security\_API | 15.2.0 |
| 2018-12 | CT#82 | CP-183109 | 0057 |  | F | Correct CAPIF\_Access\_Control\_Policy\_API | 15.2.0 |
| 2018-12 | CT#82 | CP-183109 | 0058 | 2 | F | supportedFeatures - CAPIF\_Discover\_Service\_API | 15.2.0 |
| 2018-12 | CT#82 | CP-183109 | 0059 |  | F | supportedFeatures 002 - CAPIF\_Publish\_Service\_API | 15.2.0 |
| 2018-12 | CT#82 | CP-183109 | 0060 | 1 | F | supportedFeatures 003 - CAPIF\_Events\_API | 15.2.0 |
| 2018-12 | CT#82 | CP-183109 | 0061 |  | F | supportedFeatures 004 - CAPIF\_API\_Invoker\_Management\_API | 15.2.0 |
| 2018-12 | CT#82 | CP-183109 | 0062 |  | F | supportedFeatures 005 - CAPIF\_Security\_API | 15.2.0 |
| 2018-12 | CT#82 | CP-183109 | 0063 | 2 | F | supportedFeatures - CAPIF\_Access\_Control\_Policy\_API | 15.2.0 |
| 2018-12 | CT#82 | CP-183109 | 0064 |  | F | supportedFeatures 007 - CAPIF\_Logging\_API\_Invocation\_API | 15.2.0 |
| 2018-12 | CT#82 | CP-183109 | 0065 | 2 | F | supportedFeatures - CAPIF\_Auditing\_API | 15.2.0 |
| 2018-12 | CT#82 | CP-183109 | 0067 |  | F | Redundant Editor's note | 15.2.0 |
| 2018-12 | CT#82 | CP-183109 | 0068 | 1 | F | Correct CAPIF\_API\_Invoker\_Management\_API | 15.2.0 |
| 2018-12 | CT#82 | CP-183109 | 0070 |  | F | Missing general description in A.1 | 15.2.0 |
| 2018-12 | CT#82 | CP-183109 | 0071 | 1 | F | Update mandatory error status code | 15.2.0 |
| 2018-12 | CT#82 | CP-183109 | 0072 | 3 | F | Correct resource model and add missing functions in CAPIF\_Security\_API | 15.2.0 |
| 2018-12 | CT#82 | CP-183109 | 0074 | 2 | F | Correct resource model and add missing function in AEF\_Authentication\_API | 15.2.0 |
| 2018-12 | CT#82 | CP-183109 | 0075 | 1 | F | externalDocs field in OpenAPI documents | 15.2.0 |
| 2018-12 | CT#82 | CP-183109 | 0076 | 3 | F | location header in OpenAPI documents | 15.2.0 |
| 2018-12 | CT#82 | CP-183109 | 0077 | 1 | F | version number in OpenAPI documents | 15.2.0 |
| 2018-12 | CT#82 | CP-183109 | 0078 | 2 | F | corrections to CAPIF\_Access\_Control\_Policy\_API | 15.2.0 |
| 2018-12 | CT#82 | CP-183109 | 0079 | 1 | F | corrections to CAPIF\_Logging\_API\_Invocation\_API | 15.2.0 |
| 2018-12 | CT#82 | CP-183109 | 0079 | 2 | F | Security adaptation for T8 APIs with CAPIF | 15.2.0 |
| 2018-12 | CT#82 | CP-183109 | 0080 |  | F | corrections to EventNotification | 15.2.0 |
| 2018-12 | CT#82 | CP-183109 | 0081 |  | F | corrections to theSubscriber | 15.2.0 |
| 2018-12 | CT#82 | CP-183109 | 0082 |  | F | remove 'OnboardingRequestAck' data type | 15.2.0 |
| 2019-03 | CT#83 | CP-190119 | 0083 | 1 | F | Correct GET description for retrieving service API information | 15.3.0 |
| 2019-03 | CT#83 | CP-190119 | 0084 | 1 | F | Correct PUT message for updating service APIs | 15.3.0 |
| 2019-03 | CT#83 | CP-190119 | 0085 | 2 | F | Correct AEF operations related to obtaining security info or revoking API invokers | 15.3.0 |
| 2019-03 | CT#83 | CP-190119 | 0086 | 1 | F | Correction of definition of obtaining the correct resource in Security APIs | 15.3.0 |
| 2019-03 | CT#83 | CP-190119 | 0089 | 1 | F | Correct several descriptions in clause 8 tables | 15.3.0 |
| 2019-06 | CT#84 | CP-191088 | 0090 | 1 | F | Correct CAPIF\_Logging\_API yaml file | 15.4.0 |
| 2019-06 | CT#84 | CP-191221 | 0091 | 1 | F | Copyright notice in the YAML files | 15.4.0 |
| 2019-06 | CT#84 | CP-191222 | 0092 | 1 | F | API version update | 15.4.0 |
| 2019-09 | CT#85 | CP-192158 | 0093 | 3 | F | Northbound API registeration and discovery | 16.0.0 |
| 2019-12 | CT#86 | CP-193194 | 0095 | 1 | A | Correct cardinality in event API | 16.1.0 |
| 2019-12 | CT#86 | CP-193199 | 0096 | 4 | B | Reference update: RFC 8259 | 16.1.0 |
| 2019-12 | CT#86 | CP-193199 | 0097 |  | F | Detailed information in CAPIF event notification | 16.1.0 |
| 2019-12 | CT#86 | CP-193195 | 0101 | 4 | B | Updates to Service Architecture and functional entities | 16.1.0 |
| 2019-12 | CT#86 | CP-193194 | 0103 | 1 | A | Clause reference corrections | 16.1.0 |
| 2019-12 | CT#86 | CP-193194 | 0105 | 1 | A | Conventions for Open API specification files | 16.1.0 |
| 2019-12 | CT#86 | CP-193195 | 0106 | 1 | B | Update-to-Service-Architecture | 16.1.0 |
| 2019-12 | CT#86 | CP-193195 | 0107 | 2 | B | Update-to-Service-API-Publish | 16.1.0 |
| 2019-12 | CT#86 | CP-193195 | 0108 | 1 | B | Interconnection-Service-API-Publish | 16.1.0 |
| 2019-12 | CT#86 | CP-193195 | 0109 | 2 | B | Update-to-Discover-Service-API | 16.1.0 |
| 2019-12 | CT#86 | CP-193199 | 0111 | 1 | B | Supported feature in API publish service | 16.1.0 |
| 2019-12 | CT#86 | CP-193195 | 0112 | 1 | B | API invoker details update – Service Definition | 16.1.0 |
| 2019-12 | CT#86 | CP-193195 | 0113 | 1 | B | API invoker details update – API Definition | 16.1.0 |
| 2019-12 | CT#86 | CP-193195 | 0114 | 1 | B | API Provider Registration and Update – Service Definition | 16.1.0 |
| 2019-12 | CT#86 | CP-193195 | 0115 | 3 | B | API Provider Registration and Update – API Definition | 16.1.0 |
| 2019-12 | CT#86 | CP-193195 | 0116 | 1 | B | Support for 3rd party API provider domain | 16.1.0 |
| 2019-12 | CT#86 | CP-193194 | 0118 | 1 | A | Correct the notificationDestination of ServiceSecurity object in yaml file | 16.1.0 |
| 2019-12 | CT#86 | CP-193194 | 0120 | 1 | A | Align the API name of Initiate\_Authentication | 16.1.0 |
| 2019-12 | CT#86 | CP-193212 | 0121 |  | F | Update of API version and TS version in OpenAPI file | 16.1.0 |
| 2020-03 | CT#87e | CP-200205 | 0123 | 1 | B | Published API path | 16.2.0 |
| 2020-03 | CT#87e | CP-200205 | 0124 |  | B | API Invoker Udpate – Event Updates | 16.2.0 |
| 2020-03 | CT#87e | CP-200205 | 0125 | 2 | B | API Provider Management – Open API | 16.2.0 |
| 2020-03 | CT#87e | CP-200216 | 0126 |  | F | 29.222 Rel-16 Update of OpenAPI version and TS version in externalDocs field | 16.2.0 |
| 2020-06 | CT#88e | CP-201277 | 0128 | 3 | B | Service description and operations for CAPIF\_API\_Routing\_Policy\_API | 16.3.0 |
| 2020-06 | CT#88e | CP-201277 | 0129 | 3 | B | API definition for CAPIF\_API\_Routing\_Policy\_API | 16.3.0 |
| 2020-06 | CT#88e | CP-201278 | 0130 | 3 | B | API Topology hiding | 16.3.0 |
| 2020-06 | CT#88e | CP-201230 | 0133 |  | A | Correct API publish procedure | 16.3.0 |
| 2020-06 | CT#88e | CP-201231 | 0131 | 1 | F | API Provider management API attribute name optimization | 16.3.0 |
| 2020-06 | CT#88e | CP-201231 | 0135 | 1 | F | Correct ServiceAPIDescription | 16.3.0 |
| 2020-06 | CT#88e | CP-201231 | 0136 | 2 | F | Correct service API discovery in interconnection | 16.3.0 |
| 2020-06 | CT#88e | CP-201231 | 0137 | 1 | F | Correct shareable information | 16.3.0 |
| 2020-06 | CT#88e | CP-201235 | 0138 | 1 | F | Correct the supported features in the published API | 16.3.0 |
| 2020-06 | CT#88e | CP-201235 | 0139 | 1 | F | Update general subclause for OpenAPI specification | 16.3.0 |
| 2020-06 | CT#88e | CP-201256 | 0140 | 1 | F | URI of the CAPIF APIs | 16.3.0 |
| 2020-06 | CT#88e | CP-201231 | 0141 | 1 | B | Add API category in discovery | 16.3.0 |
| 2020-06 | CT#88e | CP-201235 | 0142 |  | F | Optionality of ProblemDetails | 16.3.0 |
| 2020-06 | CT#88e | CP-201230 | 0144 | 1 | A | Clause and reference point correction | 16.3.0 |
| 2020-06 | CT#88e | CP-201231 | 0145 | 1 | F | Align interface names | 16.3.0 |
| 2020-06 | CT#88e | CP-201235 | 0146 | 1 | F | Supported headers, Resource Data type, Operation Name and yaml mapping | 16.3.0 |
| 2020-06 | CT#88e | CP-201255 | 0147 |  | F | Update of OpenAPI version and TS version in externalDocs field | 16.3.0 |
| 2020-06 | CT#88e | CP-201319 | 0149 |  | A | Required attribute corrections to CAPIF Open APIs | 16.3.0 |
| 2020-09 | CT#89e | CP-202064 | 0151 | 1 | F | Missing and inconsistent “apiVersion” notations and Location header | 16.4.0 |
| 2020-09 | CT#89e | CP-202064 | 0152 | 1 | F | CAPIF Routing Info API corrections | 16.4.0 |
| 2020-09 | CT#89e | CP-202064 | 0153 |  | F | CAPIF topology hiding correction | 16.4.0 |
| 2020-09 | CT#89e | CP-202233 | 0155 | 3 | A | Correct CAPIF security API | 16.4.0 |
| 2020-09 | CT#89e | CP-202063 | 0157 | 1 | A | Correct api invoker certificate in onboarding | 16.4.0 |
| 2020-09 | CT#89e | CP-202084 | 0158 |  | F | Update of OpenAPI version and TS version in externalDocs field | 16.4.0 |
| 2020-12 | CT#90e | CP-203139 | 0160 | 1 | F | Essential corrections and alignments | 16.5.0 |
| 2020-12 | CT#90e | CP-203126 | 0162 | 1 | A | Correct inconsistency in SecurityNotification | 16.5.0 |
| 2020-12 | CT#90e | CP-203139 | 0163 | 1 | F | Storage of YAML files in 3GPP Forge | 16.5.0 |
| 2021-03 | CT#91e | CP-210239 | 0164 | 1 | F | CAPIF\_Security API externalDocs version correction | 16.6.0 |
| 2021-06 | CT#92e | CP-211216 | 0184 |  | A | SecurityMethod data type incorrectly written some parts of the CAPIF\_Publish\_Service\_API description clause | 16.7.0 |
| 2022-03 | CT#95e | CP-220168 | 0220 |  | A | Correct inconsistencies | 16.8.0 |
| 2022-03 | CT#95e | CP-220177 | 0229 |  | F | Update of info and externalDocs fields | 16.8.0 |
| 2022-06 | CT#96 | CP-221124 | 0236 | 1 | A | Correcting the data type of the APF identifier | 16.9.0 |
| 2022-06 | CT#96 | CP-221124 | 0239 | 1 | A | Correcting the data type of the service API Identifier | 16.9.0 |
| 2022-06 | CT#96 | CP-221124 | 0242 |  | A | Correcting query parameters names in the CAPIF\_Security\_API | 16.9.0 |
| 2022-06 | CT#96 | CP-221124 | 0246 | 1 | A | Correct token request content type | 16.9.0 |
| 2022-06 | CT#96 | CP-221150 | 0249 |  | F | Update of info and externalDocs fields | 16.9.0 |
| 2022-12 | CT#98e | CP-223234 | 0261 | 2 | A | Corrections for CAPIF\_API\_Invoker\_Management\_API | 16.10.0 |
| 2022-12 | CT#98e | CP-223169 | 0269 |  | A | Corrections on Enumeration Protocol for CAPIF\_Publish\_Service\_API | 16.10.0 |
| 2022-12 | CT#98e | CP-223169 | 0272 | 1 | A | Corrections on POST request body for CAPIF\_Logging\_API\_Invocation\_API | 16.10.0 |
| 2022-12 | CT#98e | CP-223169 | 0275 |  | A | Corrections on resource URI for CAPIF\_Discover\_Service\_API | 16.10.0 |
| 2022-12 | CT#98e | CP-223169 | 0278 |  | A | Corrections on Time Range List for CAPIF\_Access\_Control\_Policy\_API | 16.10.0 |
| 2022-12 | CT#98e | CP-223236 | 0281 | 1 | A | Corrections for data type of CAPIF services | 16.10.0 |
| 2022-12 | CT#98e | CP-223187 | 0283 |  | F | Update of info and externalDocs fields | 16.10.0 |
| 2023-03 | CT#99 | CP-230135 | 0292 |  | A | Corrections to CAPIF\_Auditing\_API | 16.11.0 |