**3GPP TSG- Meeting #130 *C3-234360\_R1***

**Xiamen, China, 09th – 13th October, 2023**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.2* | | | | | | | | |
| **CHANGE REQUEST** | | | | | | | | |
|  | | | | | | | | |
|  |  | **CR** | **0317** | **rev** | **-** | **Current version:** |  |  |
|  | | | | | | | | |
| *For* [***HE******LP***](http://www.3gpp.org/3G_Specs/CRs.htm#_blank)*on using this form: comprehensive instructions can be found at* [*http://www.3gpp.org/Change-Requests*](http://www.3gpp.org/Change-Requests)*.* | | | | | | | | |
|  | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Proposed change affects:*** | UICC apps |  | ME |  | Radio Access Network |  | Core Network | **X** |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | |
| ***Title:*** | Error handling in the CAPIF layer | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Source to WG:*** |  | | | | | | | | | |
| ***Source to TSG:*** | CT3 | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Work item code:*** | NBI18 | | | | |  | ***Date:*** | | |  |
|  |  | | | |  | |  | | |  |
| ***Category:*** | **F** |  | | | | | ***Release:*** | | |  |
|  | *Use one of the following categories:* ***F*** *(correction)* ***A*** *(mirror corresponding to a change in an earlier release)* ***B*** *(addition of feature),* ***C*** *(functional modification of feature)* ***D*** *(editorial modification)*  Detailed explanations of the above categories can be found in 3GPP [TR 21.900](http://www.3gpp.org/ftp/Specs/html-info/21900.htm). | | | | | | | | *Use one of the following releases: Rel-8 (Release 8) Rel-9 (Release 9) Rel-10 (Release 10) Rel-11 (Release 11) … Rel-16 (Release 16) Rel-17 (Release 17) Rel-18 (Release 18) Rel-19 (Release 19)* | |
|  |  | | | | | | | | | |
| ***Reason for change:*** | | CAPIF layer specifies many service operations; however, the service operations do not specify any error handling. Thus, the error handling shall be added to provide harmonization across NBI API specifications. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Summary of change:*** | | The error handling is added to the service operations in the CAPIF layer. Editorial corrections. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Consequences if not approved:*** | | * Non-harmonized service operation specification across NBI APIs; * Low quality of the specification. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Clauses affected:*** | | 5.2.2.2.2, 5.3.2.2.2, 5.3.2.2.3, 5.3.2.3.2, 5.3.2.4.2, 5.3.2.5.2, 5.4.2.2.2, 5.4.2.3.2, 5.5.2.2.2, 5.5.2.3.2, 5.5.2.5.2, 5.6.2.2.2, 5.6.2.4.2, 5.8.2.2.2, 5.9.2.2.2, 5.10.2.2.2, 5.11.2.2.2, 5.11.2.3.2, 5.11.2.4.2, 5.12.2.2.2 | | | | | | | | |
|  | |  | | | | | | | | |
|  | | **Y** | **N** |  | | | |  | | |
| ***Other specs*** | |  | **X** | Other core specifications | | | | TS/TR ... CR ... | | |
| ***affected:*** | |  | **X** | Test specifications | | | | TS/TR ... CR ... | | |
| ***(show related CRs)*** | |  | **X** | O&M Specifications | | | | TS/TR ... CR ... | | |
|  | |  | | | | | | | | |
| ***Other comments:*** | | * This CR does not affect any OpenAPI file. | | | | | | | | |
| ***()*** | |  | | | | | | | | |
| ***This CR's revision history:*** | |  | | | | | | | | |

**Additional discussion(if needed):**

**Proposed changes:**

\* \* \* First Change \* \* \* \*

##### 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 clause 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; and

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;

NOTE: The {apiRoot} part of the URI structure (defined in clause 5.2.4 of 3GPP TS 29.122 [14]) for the discovered APIs can be constructed by the API invoker based on either the "domainName" attribute (which contains all the required information, e.g. FQDN or IP address, port, a deployment specific string in the form of a sequence of path segments) or the "interfaceDescriptions" attribute of the AefProfile data type.

and

3. if errors occur when processing the request, the CAPIF core function shall respond to the consumer with an appropriate error status code as defined in clause 8.1.5.

\* \* \* Next change \* \* \* \*

##### 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 clause 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; and

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 clause 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;

and

3. if errors occur when processing the request, the CAPIF core function shall respond to the API publishing function with an appropriate error status code as defined in clause 8.2.5.

\* \* \* Next change \* \* \* \*

##### 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 requesting 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 clause 8.2.2.2.3.1. For service API publishing on CAPIF-6 reference point, the requesting CAPIF core function shall also include the published API path "pubApiPath" as specified in clause 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 of the requesting CAPIF core function and received CAPIF core function identifier(s) from other CAPIF core function.

If the requesting 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 peer CAPIF core function shall:

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

2. if the requesting CAPIF core function is authorized to publish service APIs, the peer 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 peer CAPIF core function shall add its own identifier in the path; otherwise reject the HTTP POST request and skip step 3;

3. then the peer CAPIF core function shall:

a. verify the rest API Information present in the HTTP POST message and add the service APIs in the peer CAPIF core function (API registry);

b. create a new resource as specified in clause 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;

and

4. if errors occur when processing the request, the peer CAPIF core function shall respond to the peer CAPIF core function with an appropriate error status code as defined in clause 8.2.5.

\* \* \* Next change \* \* \* \*

##### 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 clause 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; and

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;

and

3. if errors occur when processing the request, the CAPIF core function shall respond to the consumer with an appropriate error status code as defined in clause 8.2.5.

\* \* \* Next change \* \* \* \*

##### 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 clause 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 clause 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;

and

3. if errors occur when processing the request, the CAPIF core function shall respond to the consumer with an appropriate error status code as defined in clause 8.2.5.

\* \* \* Next change \* \* \* \*

##### 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 clause 8.2.2.3.3.2; otherwise, if the "PatchUpdate" feature defined in clause 8.2.6 is supported, the consumer (e.g. API publishing function) may send an HTTP PATCH request message to the concerned service API resource URI in the CAPIF core function. The body of the HTTP PATCH request message shall include the requested modifications as specified in clause 8.2.2.3.3.4.

Upon receiving the above described HTTP PUT or PATCH request 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 or PATCH request message and replace/modify 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/modify 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; and

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

c. replace/modify 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;

and

3. if errors occur when processing the request, the CAPIF core function shall respond to the consumer with an appropriate error status code as defined in clause 8.2.5.

\* \* \* Next change \* \* \* \*

##### 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 clause 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 clause 8.3.2.1; and

b. return the CAPIF Resource URI in the response message;

and

3. if errors occur when processing the request, the CAPIF core function shall respond to the Subscriber with an appropriate error status code as defined in clause 8.3.5.

\* \* \* Next change \* \* \* \*

##### 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 clause 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;

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; and

3. if errors occur when processing the request, the CAPIF core function shall respond to the Subscriber with an appropriate error status code as defined in clause 8.3.5.

\* \* \* Next change \* \* \* \*

##### 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 clause 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; and

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 clause 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 clause 8.4.3; and

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;

and

3. if errors occur when processing the request, the CAPIF core function shall respond to the API invoker with an appropriate error status code as defined in clause 8.4.5.

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 7235 [9].

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

\* \* \* Next change \* \* \* \*

##### 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 clause 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;

and

3. if errors occur when processing the request, the CAPIF core function shall respond to the API invoker with an appropriate error status code as defined in clause 8.4.5.

\* \* \* Next change \* \* \* \*

##### 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 the APIInvokerEnrolmentDetails data structure with 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. Otherwise, if the "PatchUpdate" feature defined in clause 8.4.6 is supported, the consumer (e.g. API invoker function) may send an HTTP PATCH request message to the concerned service API resource URI in the CAPIF core function to modify the API invoker's profile. The body of the HTTP PATCH request message shall include the APIInvokerEnrolmentDetailsPatch data structure.

Upon receiving the above described HTTP PUT or PATCH request 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 or PATCH request 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 or PATCH 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 or PATCH 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 or PATCH request;

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 or PATCH request, with validated information; and

ii. return the updated API invoker details;

and

3. if errors occur when processing the request, the CAPIF core function shall respond to the API invoker with an appropriate error status code as defined in clause 8.4.5.

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.

\* \* \* Next change \* \* \* \*

##### 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 clause 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 clause 8.5.2.1;

4. return the security method information and the CAPIF Resource URI in the response message; and

5. if errors occur when processing the request, the CAPIF core function shall respond to the API invoker with an appropriate error status code as defined in clause 8.5.5.

\* \* \* Next change \* \* \* \*

##### 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 clause 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;

2. return the security information in the response message; and

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

3. if errors occur when processing the request, the CAPIF core function shall respond to the API invoker with an appropriate error status code as defined in clause 8.5.5.

\* \* \* Next change \* \* \* \*

##### 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 clause 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 clause 8.7.2.1; and

c. return the CAPIF Resource Identifier in the response message;

and

3. if errors occur when processing the request, the CAPIF core function shall respond to the API exposing function with an appropriate error status code as defined in clause 8.7.5.

\* \* \* Next change \* \* \* \*

##### 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 optionally a set of log query parameters to the CAPIF core function as specified in clause 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 parameters, if any; and

b. return the search results in the response message;

and

3. if errors occur when processing the request, the CAPIF core function shall respond to the API management function with an appropriate error status code as defined in clause 8.8.5.

\* \* \* Next change \* \* \* \*

##### 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 clause 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; and

3. if errors occur when processing the request, the CAPIF core function shall respond to the API exposing function with an appropriate error status code as defined in clause 8.6.5.

\* \* \* Next change \* \* \* \*

##### 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;

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; and

d. if errors occur when processing the request, the CAPIF core function shall respond to the API management function with an appropriate error status code as defined in clause 8.9.5.

\* \* \* Next change \* \* \* \*

##### 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 APIProviderEnrolmentDetails data structure that need to be updated. If the "PatchUpdate" feature defined in clause 8.9.6 is supported for modification of the API provider domain profile, the consumer (e.g. API publishing function) may send an HTTP PATCH request message to the concerned service API resource URI in the CAPIF core function. The body of the HTTP PATCH request message shall include the APIProviderEnrolmentDetailsPatch data structure.

Upon receiving the described HTTP PUT or PATCH request message:

1. the CAPIF core function shall process the updates received in the HTTP PUT or PATCH request 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/modify the representation of the resource identified by the CAPIF Resource URI of the API management function's HTTP PUT or PATCH 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; and

c. return a "200 OK" status code with the updated API provider domain information, or a "204 No Content" status code;

and

4. if errors occur when processing the request, the CAPIF core function shall respond to the API management function with an appropriate error status code as defined in clause 8.9.5.

\* \* \* Next change \* \* \* \*

##### 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;

and

3. if errors occur when processing the request, the CAPIF core function shall respond to the API management function with an appropriate error status code as defined in clause 8.9.5.

\* \* \* Next change \* \* \* \*

##### 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 clause 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; and

3. if errors occur when processing the request, the CAPIF core function shall respond to the API exposing function with an appropriate error status code as defined in clause 8.10.5.

\* \* \* End of changes \* \* \* \*