**3GPP TSG-SA WG6 Meeting #37-e S6-200776**

**e-meeting, 14th – 26th May 2020 (revision of S6-xxxxxx)**

|  |
| --- |
| *CR-Form-v12.0* |
| **CHANGE REQUEST** |
|  |
|  | **23.222** | **CR** | **0076** | **rev** | **-** | **Current version:** | **16.7.0** |  |
|  |
| *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:***  | Correction for API routing information |
|  |  |
| ***Source to WG:*** | Huawei, Hisilicon |
| ***Source to TSG:*** | S6 |
|  |  |
| ***Work item code:*** | eCAPIF |  | ***Date:*** | 2020-05-06 |
|  |  |  |  |  |
| ***Category:*** | **F** |  | ***Release:*** | Rel-16 |
|  | *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 canbe 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-12 (Release 12)**Rel-13 (Release 13)Rel-14 (Release 14)Rel-15 (Release 15)Rel-16 (Release 16)* |
|  |  |
| ***Reason for change:*** | There is confusion related to API topology hiding information and API routing information provided by CCF to AEF (which is determined as communication entry point AEF)When a service API is published and requires topology hiding, the topology hiding information is notified by CCF to AEF as per clause 8.24.2.1. This topology hiding information is used in clause 8.13.3 which enables service API invocation for topology hiding where the communication entry point AEF forwards the service API invocation to the AEF which actually provides the service API. For the procedure in clause 8.13.3 to work, the topology hiding information should consist of the service API identification information and the AEF information which provides the service API. Currently in clause 8.24.2.1, only AEF identity is specified which is not sufficient to allow procedure in clause 8.13.3 to work. Hence, AEF identity should be changed to AEF information (to include the details like IP address and port). Also the step 3 in procedure specified in clause 8.24.3 and the information element in clause 8.24.2.1 should be aligned.In another scenario for dynamic routing service API, it consider that a service API is served by multiple AEF(s) and hence requires a routing rule. The pre-condition should be corrected to clarify this aspect. For such scenarios, during service API invocation as specified in clause 8.27.3, the communication entry point AEF obtains a API routing information from the CCF and determines which AEF to forward the service API invocation. Currently, the AEF fetching of API routing information from the CCF as per clause 8.27.3 is represented as optional in the figure, which is incorrect. It should be changed to conditional. |
|  |  |
| ***Summary of change:*** | Correct the clause 8.24.2.1, 8.24.3 to correct the topology hiding information and keep it aligned.Correct the clause 8.27.3 to represent step 2 and step 3 to fetch API routing rule in solid line and modify pre-condition and the related step description. |
|  |  |
| ***Consequences if not approved:*** | The service API invocation for topology hiding and dynamic routing for service API invocation will not work. |
|  |  |
| ***Clauses affected:*** | 8.24.2.1, 8.24.3, 8.27.3 |
|  |  |
|  | **Y** | **N** |  |  |
| ***Other specs*** |  | **N** |  Other core specifications  | TS/TR ... CR ...  |
| ***affected:*** |  | **N** |  Test specifications | TS/TR ... CR ...  |
| ***(show related CRs)*** |  | **N** |  O&M Specifications | TS/TR ... CR ...  |
|  |  |
| ***Other comments:*** |  |
|  |  |
| ***This CR's revision history:*** |  |

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

#### 8.24.2.1 API topology hiding notify

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

Table 8.24.2.1-1: API topology hiding notify

|  |  |  |
| --- | --- | --- |
| Information element | Status | Description |
| Service API identification | M | The identification information of the service API with the API topology hiding |
| API exposing function information | M | Indicate the AEF which provides the service API to apply the topology hiding including the interface details (e.g. IP address, port number, URI).  |

\* \* \* Next Change \* \* \* \*

### 8.24.3 Procedure

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

Pre-condition:

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



Figure 8.24.3-1: API topology hiding via API publish

1. The API publishing function sends a service API publish request with the details of the service API to the CAPIF core function.

2. Upon receiving the service API publish request, the CAPIF core function checks whether the API publishing function is authorized to perform the service API publish. If authorized, based on the service APIs and policy, the CCF applies the topology hiding by selecting an AEF providing the topology hiding as the entry point for service API invocation. The selected AEF information is stored with the service API information received from API publish function at the CAPIF core function (API registry).

3. The CCF sends the API topology notify to the AEF selected as the entry point for service API invocation. The service API identification and the AEF information which provides the service API are included.

4. Upon receiving the notification, the AEF stores the received information for further service API invocation request forwarding.

5. The CCF sends an API publish response to the API publish function.

\* \* \* Next Change \* \* \* \*

### 8.27.3 Procedure

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

Pre-conditions:

1. The API invoker has performed the service discovery and received the details of the service API which includes the information about the service communication entry point of the AEF-1 in the CAPIF.

2. The API invoker is authenticated and authorized to use the service API.

3. The AEF-1 is the AEF acting as service communication entry point for the service API, and AEF-2 is one of the multiple destination AEF which provides the service API.



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

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

2. If the routing rule information for the service API invocation is not available, the AEF-1 sends obtain routing information request to the CAPIF core function.

3. The CAPIF core function creates routing rule information for the service API and sends obtain routing information response with the routing rule information. AEF-1 may cache the received routing rule.

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

4. The AEF-1 further resolves the actual destination of the service API address information (e.g. AEF-2) according to the routing rule information and the invocation parameters in service API invocation request.

5. The AEF-1 forwards the incoming service API invocation request to AEF-2.

6. The AEF-2 returns the service API invocation response to AEF-1.

7. The AEF-1 sends the service API invocation response to the API invoker.