**3GPP TSG-SA WG6 Meeting #52-bis-e S6-230150**

**e-meeting, 11th – 20th January 2023 (revision of S6-23xxxx)**

|  |
| --- |
| *CR-Form-v12.2* |
| **CHANGE REQUEST** |
|  |
|  |  | **CR** | **099** | **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:***  | Solve CAPIF extensibility EN |
|  |  |
| ***Source to WG:*** | Ericsson, Nokia, Nokia Shangahi Bell, Apple, Intel, Huawei, Samsung |
| ***Source to TSG:*** | SA6 |
|  |  |
| ***Work item code:*** | TEI18 |  | ***Date:*** | 2023-01-05 |
|  |  |  |  |  |
| ***Category:*** | B |  | ***Release:*** | Rel-18 |
|  | *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-16 (Release 16)Rel-17 (Release 17)Rel-18 (Release 18)Rel-19 (Release 19)* |
|  |  |
| ***Reason for change:*** | There is an EN remaining in cl.4.11.2 for protocol design requirement.Protocol extensibility does not impact CAPIF architecture and procedure. Stage 2 does not define protocol details, as long as the CAPIF APIs are extensible to accommodate different needs for non-3GPP organization (e.g. ETSI MEC), the design principle can be captured in stage 2 to guide stage 3 work. |
|  |  |
| ***Summary of change:*** | State the API design guidelines for CAPIF APIs.Add a new requirement for CAPIF extensibiliy in 4.11.2.  |
|  |  |
| ***Consequences if not approved:*** | Editor note remains. |
|  |  |
| ***Clauses affected:*** | 4.11.2, 9.1, 9.2 |
|  |  |
|  | **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's revision history:*** |  |

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*1st Change\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

### 4.11.2 Requirements

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

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

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

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

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

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*2nd Change\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## 9.1 General

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

- fundamental API guidelines, applicable to all northbound APIs utilizing CAPIF; and

- architecture design considerations, applicable to all northbound APIs utilizing CAPIF.

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

## 9.2 Fundamental API Guidelines

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

1. the function of the API;

2. the resource(s) or endpoints involved;

3. the list of supported operations and their usage;

4. the list of input and output parameters along with applicable schemas, as required;

5. the list of supported response codes;

6. the behaviour of the network entity exposing the APIs (e.g. the CAPIF core function or the API exposing function) for each supported operation;

7. the list of applicable data types; and

8. the list of applicable protocols and data serialization formats.

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

1. consistent nomenclature for the operations, data structures and resources/endpoints;

2. design principles for the use of operations for common tasks; and

3. a template for the consistent documentation of APIs.

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

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

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

2. support access control mechanisms;

3. support charging, if applicable; and

4. be backward and forward compatible with different versions of the same API.

The guidelines above are generic with regard to the API architecture. They are valid for network APIs that follow the RESTful architectural style and that expose resources towards the API invoker, as well as for network APIs of other architectures that expose general network endpoints towards the API invoker. A network endpoint represents one end of a communication channel through which the API consumer communicates with the API producer, using messages of a protocol defined by the API architecture. A resource is identified, and the corresponding endpoint is addressed, by a resource identifier (such as a URI).

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*End of Changes \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*