**3GPP TSG-CT WG4 Meeting #99eC4-204xyz**

**E-Meeting, 18th – 28th August 2020 (was C4-204083)**

|  |
| --- |
| *CR-Form-v12.0* |
| **CHANGE REQUEST** |
|  |
|  | **29.501** | **CR** | **0086** | **rev** | **1** | **Current version:** | **16.4.0** |  |
|  |
| *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:***  | Handling of JSON Arrays |
|  |  |
| ***Source to WG:*** | Ericsson, China Mobile |
| ***Source to TSG:*** | CT4 |
|  |  |
| ***Work item code:*** | SBIProtoc16 |  | ***Date:*** | 2020-07-27 |
|  |  |  |  |  |
| ***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:*** | During the design of some APIs in 5GC, some design errors have been detected, in relation to the handling of JSON arrays.It was agreed during CT4#98e to include in TS 29.501 some considerations describing those error scenarios, in order to avoid them in the design of future APIs. |
|  |  |
| ***Summary of change:*** | Add an informative annex describing typical error scenarios in APIs handling updates of JSON arrays. |
|  |  |
| ***Consequences if not approved:*** | - APIs may not be designed correctly leading to malfunction of essential network interactions between NFs.- Correction of some existing editorial issues in the formatting of clause headings |
|  |  |
| ***Clauses affected:*** | 4.4.3, 4.5, 4.5.1, Annex X (new) |
|  |  |
|  | **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:*** |  |

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

### 4.4.3 Callback URI structure

The purpose of the callback URI is to enable NF service consumer to provide the URI to be used by an NF Service Producer to send notification or callback requests.

The callback URI shall be in the form of an absolute URI as defined in clause 4.3 of IETF RFC 3986 [9], including an authority, and excluding any query component, any fragment component and any userinfo subcomponent.

Therefore, callback URI consists of the following components, specified with ABNF syntax (see IETF RFC 5234 [26]):

URI = scheme ":" "//" host [ ":" port ] / path

Where 'host' is either an FQDN or an IP address and the 'path' is a path to an NF consumer resource.

## 4.5 Resource Representation and Content Format Negotiation

### 4.5.1 Resource Representation

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

Annex X (Informative):
Considerations for handling of JSON arrays

This Annex provides guidelines on the use of JSON arrays on the SBI APIs of the 5GC.

As described in clause 5.2.4, the data model of an API definition in a 5GC API consists of any of the different data types supported by OpenAPI, corresponding to the different data structures found in the JSON data format. One of these structures is the "array", representing a set of ordered values.

It should be noted that, while JSON allows that the value of the elements of an array may be of different types, in OpenAPI 3.0.0 this is further restricted, and all elements of an array shall be of the same type.

Also, it is important to note that the JSON format itself, as specified in IETF RFC 8259, does not define any syntax to refer to specific array elements.

However, there are certain conventions to specify mechanims to refer to array elements, e.g. based on the position a given element has in the array.

In particular, the JSON Pointer syntax defines a string syntax for identifying a specific value within a JSON document. This syntax consists of a number of tokens separated by the "/" character; in order to refer to a specific element in an array, then token shall contain an unsigned decimal value, indicating the zero-based index of the element in the array.

EXAMPLE:

JSON document:

{

 "attr1": 0,

 "attr2": true,

 "attr3": [ 1, 2, 3 ]

}

JSON Pointer expression "/attr3/0" evaluated on such JSON document: 1

There are several scenarios, frequently employed in the 5GC APIs, that make use of the JSON Pointer mechanism to refer to specific elements in an array:

- Update of resources using the PATCH method, as described in clause 4.6.1.1.3.2. If the syntax used in the PATCH request payload is based on the "JSON Patch" format, then the JSON Pointer mechanism is used to specify patch operations applied to specific array elements.

- Notifications of events (such as data changes), as described in clause 4.6.2.3, using as notification payload the notation defined in "NotifyItem" / "ChangeItem" data types defined in 3GPP TS 29.571 [xx]. This notation is similar to "JSON Patch", so it also makes use of the JSON Pointer syntax to refer to specific array elements.

- Explicit usage of attributes containing JSON Pointer expressions in request or response payloads.

In these scenarios, it is critical that any JSON Pointer expression is applied by both client and server on the exact same array representation, since otherwise the indexes may vary, and the JSON Pointer will give unexpected results.

A typical scenario that may create issues could be as follows:

1. NF Service Consumer sends a first GET request towards a NF Service Producer, including certain query parameters in the HTTP request, that result into retrieving a resource representation that contains a subset of the data that the NF Service Producer holds under such resource. When such subset refers to returning just some of the elements of an array, rather than the entire array, then the content of the array will differ between consumer and producer.

2a. The NF Service Consumer sends a subsequent PATCH request towards the NF Service Producer, with the intention to modify a given element of the array (specified by the array index, per the JSON Pointer syntax). This results into the server modifying a wrong element in the array, given that the NF Service Producer contains a different array.

2b. Alternatively, the NF Service Consumer may subscribe to be notified by the NF Service Producer when a given resource representation has changed. When the NF Service Producer detects such a change, it sends a notification that may include a reference to an array index, which may be different than the array index kept by the NF Service Producer.

Another scenario that may lead to incorrect array updates is:

1. Two different NF Service Consumer sends a GET request towards a NF Service Producer to retrieve the representation of a certain resource.

2. A first NF Service Consumer NF1 sends a PATCH request towards the NF Service Producer, with the intention to delete a given element of the array (specified by the array index, per the JSON Pointer syntax). This results in some of the array indexes being changed (of those elements placed after the deleted element)

3. A second NF Service Consumer NF2, sends another PATCH request towards the NF Service Producer, to modify any of the elements of the array whose index was affected by the previous operation done by NF1. This results in modifying unintendedly the wrong array element.

The design of 5GC SBI APIs should take into account these scenarios and provide mechanisms to avoid unintended modifications of array elements, when they are referred by their position index in the array.

To achieve these, both NF Service Consumer and Producer (taking the role of HTTP client and server) should ensure that any resource update takes place on a known and current resource representation, based on the content of ETag values sent along with resource representations by the resource owner.

If an NF Service Consumer needs to cache a resource representation received from an NF Service Producer (i.e. the JSON information received in an HTTP response message), it shall use exact the same representation of arrays as received from the service producer.

When sending notifications or modifications whose semantics is based on the JSON Pointer syntax, the sending NF shall use exact the same representation of arrays as previously signalled in a previous interaction with the received of such notifications/mondifiations.

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