**3GPP TSG-SA3 Meeting #115 *S3-240852-r3***

**Athens, Greece, 26 February -01 March 2024 *merger of S3‑240426,*** ***S3‑240793 and S3‑240636***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.1* | | | | | | | | |
| **CHANGE REQUEST** | | | | | | | | |
|  | | | | | | | | |
|  | **33.122** | **CR** | **0060** | **rev** | **1** | **Current version:** | **18.2.0** |  |
|  | | | | | | | | |
| *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 | **X** | Radio Access Network |  | Core Network | **X** |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | |
| ***Title:*** | Access token details | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Source to WG:*** | Huawei, HiSilicon, Ericsson, Xiaomi | | | | | | | | | |
| ***Source to TSG:*** | S3 | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Work item code:*** | SNAAPPY | | | | |  | ***Date:*** | | | 2024-02-18 |
| ***30*** |  | | | |  | |  | | |  |
| ***Category:*** | **F** |  | | | | | ***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 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-15 (Release 15) Rel-16 (Release 16) Rel-17 (Release 17) Rel-18 (Release 18)* | |
|  |  | | | | | | | | | |
| ***Reason for change:*** | | To address the Editor’s Note, i.e. clarification between access tokens used for existing CAPIF implementations and access tokens used for northbound CAPIF implementations. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Summary of change:*** | | Provide text for clarification | | | | | | | | |
|  | |  | | | | | | | | |
| ***Consequences if not approved:*** | | Misleading on which token to use | | | | | | | | |
|  | |  | | | | | | | | |
| ***Clauses affected:*** | | Annex C, 6.5.3.1, 6.5.3.2, 6.5.3.3, | | | | | | | | |
|  | |  | | | | | | | | |
|  | | **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:*** | |  | | | | | | | | |

**\*\*\* START OF 1st CHANGES \*\*\***

Annex C (normative):  
Access token profile

# C.1 General

The information in this annex provides a description of two types of access tokens, i.e., the access token used in the ‘Method 3 – TLS with OAuth token’ authentication and authorisation method (i.e. used for existing CAPIF implementations, see clause 6.5.2.3) and access token used in RNAA (see clause 6.5.3). Characterization of the access token, how to obtain the access token, how to validate the access token, and how to refresh the access token is explained.

An ‘Method 3 – TLS with OAuth token’ access token or an access token used in RNAA has the following chanracterics:

- Shall be encrypted when transported over the CAPIF 1/1e and CAPIF 2/2e interfaces (e.g. using TLS);

- Shall be a bearer type as specified in IETF RFC 6750 [5];

- Shall be encoded as a JSON Web Token as specified in IETF RFC 7519 [6];

- Shall be protected by the JSON signature profile as specified in IETF RFC 7515 [7]; and,

- Shall be validated per OAuth 2.0 [4], IETF RFC 7519 [6] and IETF RFC 7515 [7].

# C.2 Access token profile

## C.2.1 General

The ‘Method–3 - TLS with OAuth token’ access token or an access token used in RNAA contains the token claims described in C.2.2. Token claims of both types’ tokens are provided by the CAPIF Core Function and contain authentication and authorization information about the API Invoker. Token claims are used by the API Exposing Function for authorization of API Invoker northbound API requests.

## C.2.2 Token claims

The CAPIF ‘Method–3 - TLS with OAuth token’ access token or an access token used in RNAA shall convey the following claims as defined in IETF RFC 7519 [6] and IETF RFC 6749 [4].

Table C.2.2-1: Access token standard claims

|  |  |
| --- | --- |
| Parameter | Description |
| exp | REQUIRED. The expiration time of the access token. Implementers MAY provide for some small leeway, usually no more than a few minutes, to account for clock skew (not to exceed 30 seconds). |
| client\_id | REQUIRED. The identifier of the API Invoker making the API request as previously established with the CAPIF Core Function through onboarding. |
| scope | REQUIRED. A string containing a space-delimited list, comprising of the following as scopes associated with this token:  - List of Services per AEF (e.g. “AEF1:Service1,Service2,Service3,...,ServiceX;  AEF2:Service1,Service2,Service3,...,ServiceZ”) |

The CAPIF OAuth access token shall additionally convey the following claim for RNAA.

Table C.2.2-X: Access token customized claims

|  |  |
| --- | --- |
| Parameter | Description |
| resOwnerId | OPTIONAL. Resource owner ID. |

The ‘exp’and ‘scope’ parameters of the access token shall be determined by the CAPIF core function based upon the client\_id of the API Invoker provided in the Access Token Request message.

The scope parameter ‘List of Services per AEF’ shall contain a full or partial list of services which the API Invoker is permitted to access at each AEF.

# C.3 Obtaining tokens

## C.3.1 General

Once an API Invoker has successfully performed onboarding with the CAPIF Core Function, the API Invoker may request the CAPIF ‘Method–3 - TLS with OAuth token’ access tokens using ‘Method 3 – TLS with OAuth token’ defined in clause 6.5.2.3 or request access tokens used in RNAA using the methods defined in clause 6.5.3. Figure C.3.1-1 shows the access token request and access token response message exchange.



Figure C.3.1-1: Requesting an access token

NOTE 1: Implementation of the OAuth 2.0 token and authorization endpoints within the CAPIF Core Function are out of scope of this document.

NOTE 2: As described in IETF RFC 6749 [4] clause 4.4, for the CAPIF ‘Method–3 - TLS with OAuth token’ access tokens, the client authentication is used as the authorization grant, therefore no additional authorization request is needed.

## C.3.2 Access token request

To obtain an access token, the API Invoker makes a request to the CAPIF Core Function by sending an Access Token Request message with the following parameters using the "application/x-www-form-urlencoded" format, with a character encoding of UTF-8 in the HTTP request entity-body. The access token request parameters are shown in table C.3.2-1.

Table C.3.2-1: Access token request message parameters

|  |  |
| --- | --- |
| Parameter | Values |
| grant\_type | REQUIRED. The value shall be set to "client\_credentials or “authorization\_code”". |
| client\_id | REQUIRED. The identifier of the API Invoker making the request. It shall match the value that was assigned to the API Invoker during the onboarding process. |
| client\_cred | OPTIONAL. The client credential that was provided to the API Invoker during the onboarding process. |
| Redirect\_uri | OPTIONAL. The value shall be identical with the value in authorization request once authorization code grant or PKCE is used. |
| code | OPTIONAL. The authorization code received from the CCF for RNAA once authorization code grant or PKCE is used. |
| code\_verifier | OPTIONAL. If the authorization code grant with PKCE flow is selected, the code verifier is used by the CCF to check the code\_challenge according to IETF RFC 7636 [11] once PKCE is used. |
| scope | OPTIONAL. A string containing a space-delimited list, comprising of the following as scopes associated with this token:  - List of Services per AEF (e.g. “AEF1:Service1,Service2,Service3,...,ServiceX;  AEF2:Service1,Service2,Service3,...,ServiceZ”) |

If the token is used for RNAA (see clause 6.5.3), the parameter resOwnerID is used for the resource owner ID.

|  |  |
| --- | --- |
| resOwnerID | OPTIONAL. Resource owner ID |

**\*\*\* END OF 1st CHANGES \*\*\***

**\*\*\* START OF 2nd CHANGES \*\*\***

#### 6.5.3.1 General

The authorization function shall obtain the necessary permission from the resource owner for allowing the API invoker to access a northbound API.

RNAA shall use token-based authorization using OAuth 2.0 framework with the following roles:

- The API invoker has the role of the OAuth 2.0 client.

- The CCF has the role of the OAuth 2.0 authorization server, i.e., providing the access token used for RNAA.

- The AEF has the role of the resource server.

The access tokens used for RNAA shall contain the resource owner ID.

The resource owner, but the resource owner ID is specified as the GPSI of the corresponding UE if the resource is related to a UE.

NOTE: The present document does not specify the resource owner.

The access token shall include the resource owner ID (i.e *resOwnerId* claim) and the API invoker ID. The resource owner ID is GPSI. The API invoker ID binds the token to the API invoker. To avoid privacy issues, GPSI should be different from MSISDN, SUPI etc.

The AEF shall check if the token includes *resOwnerId* claim, which includes resource owner ID, to identify that it is a token used in RNAA.

AEF shall do the authorization check of the API invocation request for accessing the resources of the resource owner. AEF checks the request against the token, including

1) checking the token integrity and

2) checking whether the GPSI (if present) in the API invocation request is compliant with the resource owner ID in the access token. As the token includes resource owner ID, there is no need for additional UE authentication in API invocation. Moreover, the token should be able to restrict the API invoker to a specific resource (e.g., location, QoS, PDN connectivity status) of the resource owner.

For OAuth flows involving redirection, authentication between CCF/AUF and UE should be performed after API Invoker redirects the UE to CCF/AUF.

In case of an external AF (i.e., not the application on the UE) being the API invoker, for mutual authentication of API invoker AF and API exposing function, the authentication methods of clause 6.4 and clause 6.5.2 are reused.

For authorization, the following flows may be used:

- Client credential flow (according to RFC 6749 [4]),

- Authorization code flow (according to RFC 6749 [4]), or

- Authorization code flow with PKCE (according to RFC 7636 [11]).

CCF shall indicate the supported flows to the API invoker.

CCF shall give service authorization which subscribers or users can use RNAA.

NOTE: In tthe present document, only a UE accessing its own resources is considered if the API invoker is on a UE.

**\*\*\* END OF 2nd CHANGES \*\*\***

**\*\*\* START OF 3rd CHANGES \*\*\***

#### 6.5.3.2 Authorization using oauth client credential flow

If client credential flow is used for authorization of the API invoker by the AEF, the procedures in RFC 6749 [4] shall be followed with the following profile:

- The access token request message may include the resource owner ID.

NOTE 1: If the API invoker is on a UE, the CCF obtains its GPSI during authentication.

Editor’s note: the mapping of API Invoker ID and GPSI is left for stage 3.

- The CCF shall check whether the API invoker is entitled to consume the API and allowed to access the resources of the resource owner, by using authorization information available in the CCF.

- If the API invoker is on a UE, the CCF shall check that the UE is accessing its own resources. If the API invoker is an AF not on a UE, the check is omitted.

NOTE 2: How to get the authorization from the resource owner and store it in the CCF is out of scope of the present document.

**\*\*\* END OF 3rd CHANGES \*\*\***

**\*\*\* START OF 4th CHANGES \*\*\***

#### 6.5.3.3 Authorization using authorization code (optional PKCE) flow

If authorization code flow, optionally with PKCE, is used by the AEF for authorization of the API invoker, the procedures in RFC 6749 [4] and optionally RFC 7636 [11] shall be followed, with the following profile:

- The authorization token and/or authorization request may include the resource owner ID.

Editor’s Note: Whether and how the token and/or authorization request can include resource owner ID is left to stage 3.

NOTE: If the API invoker is on a UE, the CCF obtains its GPSI during authentication.

Editor's note: the mapping of API Invoker ID and GPSI is left for stage 3.

- The resource owner dynamically authorizes the API invoker to access the resource owner's resources as described in RFC 6749 [4] and optionally RFC 7636 [11].

- If the API invoker is on a UE, the CCF shall check that the UE is accessing its own resources. The access token shall contain the resource owner ID (i.e. GPSI) and the API invoker ID. If the API invoker is an AF not on a UE, the check is omitted.

**\*\*\* END OF 4th CHANGES \*\*\***