**3GPP TSG-SA3 Meeting #121 S3-25xxxx**

**Gothenburg, SWEDEN, 7 - 11 April 2025**

**Source: <Your COMPANY NAME>**

**Title: Pseudo-CR on <Document TITLE>**

**Document for: Approval**

**Agenda item: x.x**

**Spec: 3GPP TS/TR <TS/TR number>**

**Version: <TS version>**

**Work Item: <Work Item>**

**Comments**

pCR to draft CR S3-251114 from SA3#120

so far including comments

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

### 6.5.3 Authentication and authorization for RNAA

#### 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.

The authorization function shall support finer level authorization as specified in TS 23.222 [X].

The ROF may support finer level authorization as specified in TS 23.222 [X].

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 may be the user of the UE or the owner of the subscription depending on the use case and regulations. 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 and the API invoker ID. The resource owner ID is the 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 2.0 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 OAuth 2.0 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 selected flows to the API invoker.

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

For selecting the authorization method, the procedure as specified in clause 6.3.1.2 is used with the following RNAA specific additions. The API invoker shall include in the Security Method Request the supported RNAA authorization flows. The CCF shall determine the RNAA authorization flow based on the RNAA capabilities of the CCF, AEF, and API invoker. The API invoker shall use the determined RNAA authorization flow in the subsequent communication with the CCF and AEF.

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

#### 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 and details on finer level authorization described in clause 6.Y.

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.

#### 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. It may also include finer level authorization information described in clause 6.Y.

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].

- The CCF shall check the finer level authorization if included in the request message. 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.

#### 6.5.3.4 Revocation

The CCF can initiate the Authorization Revocation Request message as defined in clause 8.23.4 of TS 23.222 [3] with additional information to identify the RNAA-related revoked token. The additional information may include information for finer level authorization.

NOTE: The CCF can receive a revocation request message from the resource owner via the UE, resource owner function, web page etc. All these mechanisms are out of the scope of the present document.

AEF, storing the information about the RNAA-related revoked token, shall check whether the token presented by an API invoker is revoked or not, before responding to the API invoker’s invocation request.

The CCF provided notification message to the API invoker shall include the information to identify the RNAA-related revoked token.

AJ: I suggest to split here. Maybe we can agree on the above in April (and do not touch this part of the CR anymore?)

Step-wise approach!

Below would then be a DRAFT CR to finalize in May meeting – open issue feature level?

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

## 6.Y Authorization for finer level service API access

Editor’s Note: Details to be done.

For the finer level service API access, the authorization request via CAPIF-2 includes the indication of the requested service at the respective granularity. For RNAA, the request also includes the GPSI of the UE.

The CCF verifies the API invoker ID and the requested service operation/resource, if available. The authorization result is sent to the API invoker if the verification is successful. The API Invoker includes the required additional information to CCF in the Access token Request.

NOTE 1: A finer level granularity can include: Resource Owner ID, operations (e.g. retrieve, create, etc), features (e.g. feature 1, feature 2, etc) and resources (e.g. resource 1, resource 2, etc).

When the verification is completed, the CCF will include the authorization details, with the new granularity, into the access token returned to the API Invoker.

The previously provided access token will allow the AEF to correctly authorize, or deny, the request by the mechanism already available to AEF. AEF shall be able to verify the details for finer granularity access control.

Comment: we need a clear definition of finer level, see below excerpts

Service level: limit a service by only allowing some service operations



Resource level: limit to only use POST or GET???

Limit to resource and to the custom operation



Feature level: e.g. location (finer level could be e.g. to limited only high level location exposed and not a detailed location)

Do we have the same understanding in stage2 & stage 3?

??? Should we have also limiting to allow a subset of the services?

??? how to handle feature level?

Stage 3: understanding as finer level of HTTP msg as put/get/post/delete which is per service operation Stage 2: subset of service operations / a services has several service operations

Resource in CT3 ?? they call URI a resource.

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