**3GPP TSG-SA3 Meeting #109AdHoc-e *draft\_S3-230230-r2***

**Electronic meeting, 16 - 20 January 2023**

**Source: Ericsson**

**Title: A solution for authorization before allowing access to resources**

**Document for: Approval**

**Agenda Item: 5.11**

# 1 Decision/action requested

***It is proposed to approve the pCR to TR 33.884.***

# 2 References

[1] 3GPP TR 33.884: " Study on application enablement aspects for subscriber-aware northbound API access"

# 3 Rationale

One of the main objectives of SNAAPP security study is to investigate possible solutions to address the requirement about authorization by the resource owner before allowing access to resources of the resource owner.

This contribution proposes a solution to address this requirement. The solution utilizes existing mechanisms as much as possible.

# 4 Detailed proposal

Approve the following changes to TR 33.884 [1].

\*\*\* Start of Change \*\*\*

## 6.0 Mapping of solutions to key issues

Table 6.0-1: Mapping of solutions to key issues

|  |  |  |  |
| --- | --- | --- | --- |
| Solutions | KI#1 | KI#2 |  |
| Solution #3: UE Originated API invocation using OAuth Client Credential Grant | x |  |  |
| Solution #X: Authorization before allowing access to resources |  | x |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

## 6.X Solution #X: Authorization before allowing access to resources

### 6.X.1 Introduction

This solution addresses the security requirement about authorization by the resource owner before allowing access to resources of the resource owner, which is detailed in key issue #2.

It is assumed that authorization information by the resource owner has been received and stored in the authorization server with a method out of the scope of this solution.

Also, the consideration whether the resource owner is the subscription user or the subscription owner is out of scope of this solution.

The MNO learns the authorization information from the subscription user or from the subscription owner and stores the authorization information, which is bound to the UE identifier, in the PLMN trusted domain. How the MNO authenticates the resource owner and learns the authorization information is out of scope of this solution.

This solution covers the case that the API invoker is the AF accessing to resources related to a UE or the API invoker is the application in the UE accessing to resources related to that UE.

How the AF maps the target username in the application layer into the UE identifier is out of scope of this solution. The authentication and authorization behind the AF-CAPIF interaction for the triggering UE and user is out of scope.

This solution assumes that the authorization server is co-located with the CAPIF Core Function (CCF). This solution does not specify the place where the authorization information is stored. The CCF may store the authorization information in an external storage, and in this case it is assumed that there is a secure channel between the CCF and the external storage.

Editor’s Note: Clarification of storage is FFS.

Editor’s Note: Consideration of user cases if FFS.

### 6.X.2 Solution details

Below describes the steps of the procedure for "authorization before allowing access to resources", which is shown in Figure 6.X.2-1.



Figure 6.X.2-1: Authorization before allowing access to resources

How the authentication is executed for the API invoker that runs in the UE is out of scope of this solution. In general, the solution doesn’t focus on the authentication of the API invoker.

1. The API invoker and the CCF execute authentication procedures and establish a secure channel as specified in TS 33.122 [5].
2. The API invoker request OAuth access token as specified in TS 33.122 [5].
3. The CCF verifies the request.

Steps 4-6 is executed if resource owner authorization check is needed for the API invocation.

1. The CCF may fetch resource owner authorization information from the storage.
2. The storage sends the resource owner authorization information.
3. The CCF issues an access token that includes an indication for the resource owner authorization. The CCF sends the issued token to the API invoker.
4. The API invoker and API exposing function establish a secure channel.
5. The API invoker sends the token to the API Exposing Function.
6. The API Exposing Function verifies the token and checks the resource owner authorization information before allowing access to the resources related to the subscription.
7. The API Exposing Function returns the API invocation response to the API invoker.

For the UE originated API invocation case where an application triggers the API invocation by the application in the operating system in the UE, this solution assumes that an authorization in the granularity of application level is executed by a mechanism, like allowing users in the mobile phones to control the permission of application to access resources such as microphone of the mobile phone, provided by the operating system, which is out of scope of this solution. To give the access control power to the MNO considering the permission from the user or subscriber, the MNO needs to retrieve the permission/authorization information from the subscriber or user and store it in the authorization server/storage with an out of scope mechanism. This solution applies to the specific case where the application is accessing to the resources of the UE on which the application is running. The case of accessing resource of other UEs by the UE is not covered in this solution.The following procedure, depicted in Figure 6.X.2-2, shows how the case that the resource owner revokes the authorization after the CCF issues a token can be handled.



Figure 6.X.2-1: Revocation of resource owner authorization

1. Resource owner authorization is revoked.
2. The CCF informs the AEF about the revocation. (It is assumed that the AEF has subscribed to the CCF event exposure service).
3. The API invoker and AEF executes some authentication and establish a secure channel using TLS.
4. The API invoker sends the access token in the NB API call.
5. The AEF verifies the access token, checks the resource owner authorization. Since the resource owner authorization has been revoked, the AEF rejects the request.
6. The AEF sends the rejection response to the API invoker.

### 6.X.3 Evaluation

The solution utilizes existing mechanisms as much as possible.

The solution addresses the following cases:

* The AF accesses resources related to a UE
* The application in the UE is accessing the resources related to that UE.

The solution assumes that there is a mechanism in the UE for authorization in the application-level granularity.

This solution is a future proof solution considering possible extension of the definition of resource owner to cover the subscribers in addition to the users.

This solution assumes that the API invoker application in the operating system of the UE is authenticated and authorized by a method out of scope.

Editor’s Note: Further evaluation is FFS.

\*\*\* End of Change \*\*\*