**3GPP TSG-SA5 Meeting #158 *S5-247135d3***

**, , - Revision of S5-246519**

**Source: Ericsson España S.A.**

**Title: pCR TR 28.879 Authorization UC - updates**

**Document for: Approval**

**Agenda Item: 6.19.21**

# 1 Decision/action requested

***The group is asked to discuss and approve the proposal.***

# 2 References

[1] 3GPP TR 28.879: " Study on OAM for service management and exposure to external consumers".

# 3 Rationale

This pCR aims to update the authorization UC, by providing further details on:

- Use case description, clarifying that the authorization is a two-stage process that requires interaction over CAPIF-1e (Authorization request) and CAPIF-2e interface (Service API invocation)

- Solutions and evaluations, specifying the touchpoints of CAPIF authorization framework with MSAC.

# 4 Detailed proposal

It is proposed to make the following changes in the latest version of TR 28.879 [1].

|  |
| --- |
| **Begin Change** |

### 5.1.4 Use case #4: Authorization of the external MnS consumer to access the management service API

#### 5.1.4.1 Description

Upon completion of discovery, the API invoker is now ready to consume service APIs. To gain access to one or more service APIs, the API invoker needs to get authorized using CAPIF built-in OAuth2.0 framework. As noted in clause 6.5.2.3 of TS 33.122 [14], the authorization use case is a two-stage process, as follows:

* Stage #1: Authorization request (over CAPIF-1e interface). In this stage, the API invoker requests the CCF for authorization to access one or more service APIs. The CCF issues this authorization to the API invoker in the form of an JWT access token. This access token contains the API invoker’s permissions for requested service APIs.
* Stage #2: Service API invocation (over CAPIF-2e interface). In this stage, the API invoker issues a Service API invocation request to the AEF, sending the URI of the service API along with the access token received from the stage #1. The request is subjected to authorization, checking the API invoker’s service API invocation against the permissions in the access token. If authorization is successful, the requested service API can be invoked and the appropriate response is returned to the API invoker.

The 3GPP management system allows configuring authorization information (permissions) on a per MnS consumer basis, for authorization, leveraging role-based access control (RBAC). TS 28.319 [29] defines a framework for such a capability set, which is referred to as management service access control (MSAC). MSAC information is used by the MnS producer to authorize incoming MnS consumer requests.

When using CAPIF as the framework to expose MnS, the external MnS consumer plays the role of the API Invoker. Putting the above into the context, the following can be noted:

* The authorization information of an external MnS consumer is defined using MSAC.
* The defined authorization information of an external MnS consumer is put into an access token.
* An access token is issued by the CCF (sends the access token to the external MnS consumer) and interpreted by the MnS producer (reads the access token to authorize external MnS consumer’s service API invocation request).

The issue here is how to ensure CCF is able to issue access tokens that can be understood/interpreted by the MnS producer. This requires that CCF has access to external MnS consumer authorization information.

#### 5.1.4.2 Potential requirements

**PREQ-FS\_MExpo-Auth-0x:** The 3GPP management system shall provide the capability to define authorization information for an external MnS consumer using MSAC.

**PREQ-FS\_MExpo-Auth-0y:** The 3GPP management system shall provide the capability to make external MnS consumer’s authorization information available to the CCF, so that CCF can grant authorization for an external MnS consumer.

#### 5.1.4.3 Potential solutions

#### 5.1.4.3.1 Potential solution #1: Using Identity class to definethe authorization information of an external MnS consumer.

5.1.4.3.1.1 Introduction

The information model for MSAC is described in clause 7 of TS 28.319 [29]. The MSAC information model (see clause 7.3 of TS 28.319 [29]) specifies three classes: Identity, which represents an identity of a MnS consumer, and the associated roles; Role, which represents the role name and a list of permissions in a network management system; and AccessRule, which represents a permission in a network management system.

This potential solution proposes using Identity to define the authorization information of an external MnS consumer.

5.1.4.3.1.2 Description

During the API invoker enrolment stage, a subscription for the external MnS consumer is created. This subscription defines the list of published service APIs that the external MnS consumer can discover and access later, together with SLA related to API invocations (e.g., quota, throttling). This subscription has the information needed to define the authorization information for an external MnS consumer. To that end, the following occurs:

1. CAPIF administrator identifies the API provider domain(s) where enrolled service APIs belongs to; in this solution, the API provider domain is MSED.
2. CAPIF administrator requests the administrator(s) of identified API provider domain(s) to define the authorization information for an external MnS consumer, using the access control framework applicable in the domain(s); in this solution, the applicable access control framework is MSAC.
3. The administrator uses MSAC to create an Identity instance, by associating it to one or more Role instances, each listing one or more AccessRule instances. This information is stored in the authentication and authorization service producer (see clause 4.9 of TS 28.533 [2]).
4. The authentication and authorization MnS producer generates an onboarding credential for the external MnS consumer.
5. The authentication and authorization MnS producer associates the Identity instance created in step 3 with the onboarding credential created in step 4. This association is kept in the authentication and authorization service producer, i.e. not disclosed to CCF nor external MnS consumer.
6. The onboarding credential created in step 4 together with CCF details (address, root CA certificate) are sent to the external MnS consumer, so that it can initiate the onboarding.

#### 5.1.4.3.x Potential solution #x: Identity class made available to the CCF

5.1.4.3.x.1 Introduction

From the 3GPP management system perspective, the Identity class represents the only MSAC information that can be made available to any access control system. On the other hand, AccessRule class attributes contains 3GPP management specific information (e.g., JEX expressions, DNs of MOIs, etc) that is not understood/interpretable by the CCF. On the other hand, the Role class includes a list of access rules based on AccessRule class, so not useful neither for CCF.

In this regard, the MSAC information that is eligible for CCF access is limited to the Identity class. This potential solution focuses on how to make an Identity instance available to the CCF, so that the CCF can use this information to grant authorization to external MnS consumers.

5.1.4.3.x.2 Description

The workflow describing the solution is depicted in Figure 5.1.4.3.x.1-1. The pre-condition requires executing the steps described in potential solution #1 (see clause 5.1.4.3.1.2), which is where the Identity instance for the API invoker is created.

 

Figure 5.1.4.3.x.1-1: Solution #x workflow.

The steps 1-9 corresponding to the API invoker onboarding procedure.

1. To begin the onboarding procedure, the API invoker establishes a secure connection with the CCF based on TLS server-side authentication. The server certificate is CCF’s Root CA, which was sent to the API invoker after enrolment (see step 6 in clause 5.1.4.3.1.2).
2. The API invoker sends an onboard API invoker request to the CCF over the CAPIF-1/CAPIF-1e interface. This request involves providing the onboarding enrolment information using the “APIInvokerEnrolmentDetails” data type (see clause 8.4.4.2.2 of TS 29.222[13]). This data type includes the onboarding credential, which was sent to the API invoker after enrolment (see step 6 in clause 5.1.4.3.1.2).
3. The CCF sends an acknowledgment for receiving the onboard API invoker request to the API invoker.
4. The CCF takes the “onboarding credential” from the APIInvokerEnrolmentDetails data type, and sends it to the authentication and authorization MnS producer
5. The authentication and authorization MnS producer validates the received “onboarding credential”.
6. If the “onboarding credential” is valid, the authentication and authorization MnS producer retrieves the MSAC Identity associated to the onboarding credential (see step 4 of clause 5.1.4.3.1.2). The retrieved MSAC Identity instance represents the authorization information of the API invoker.
7. The authentication and authorization MnS producer sends the associated MSAC identity for the API invoker to the CCF.
8. Upon receiving the associated MSAC identity, the CCF generates the “API invoker id” that represents a unique identifier for the API invoker in the CCF. The CCF associates the generated “API invoker id” with the received MSAC identity.
9. The CCF sends an onboard API invoker response to the API invoker with the response body represented by the “APIInvokerEnrolmentDetails” data type. The response includes the assigned “API invoker id”

To gain access to one or more service APIs, the API invoker needs to get authorized using CAPIF built-in OAuth2.0 framework. As described in clause 5.1.4.1, this authorization procedure is a two-stage process, i.e., Authorization request (steps 10 - 16) and Service API invocation.

1. When the API invoker wants to invoke specific service APIs, it establishes a secure connection with the CCF based on TLS mutual authentication.
2. The API invoker sends an access token request to the CCF to invoke specific service API(s), providing the “API invoker id” and optionally a list of service API(s) it wants to invoke. The request body carries the information described by the “AccessTokenReq” data type (see Table 5.1.4.3.x.2-1).
3. The CCF validates the request, and if valid, retrieves the MSAC Identity associated to the “API-invoker-id” (see step 9). This “API-invoker-id” is in the “client\_id” parameter of the received access token request.
4. The CCF sends access token request to the authentication and authorization MnS producer. In this request, the “client\_id” parameter in the access token request is set to the retrieved MSAC Identity.
5. The authentication and authorization MnS producer validates the request and generates an access token. This token will contain, as part of token claims, the allowed APIs (MnSes) that the API invoker is authorized to invoke.
6. The authentication and authorization MnS producer sends the generated access token to the CCF.
7. Upon receiving the access token, the CCF forwards the received access token response to the API invoker. The CCF sends a "service API authorization response", which carries the information described by the “AccessTokenRsp” data type (see Table 5.1.4.3.x.2-2)

Finally, the API invoker can successfully invoke the service API at the AEF as described in clause 6.5.2.3 of TS 33.122[14].

NOTE: How the CCF interacts with authentication and authorization MnS producer is up to implementation and not subject to standardization. The implementation may choose to collocate or not the CCF and the authentication and authorization MnS producer.

Table 5.1.4.3.x.2-1: Definition of AccessTokenReq data type.

| Attribute name | Attribute additional information | Comments |
| --- | --- | --- |
| grant\_type | The data type of this attribute is defined as "string" and presence qualifier is defined as "M" (see table 8.5.4.2.6-1 of TS 29.222 [13]). | Set to “client\_credentials” |
| client\_id | The data type of this attribute is defined as "string" and presence qualifier is defined as "M" (see table 8.5.4.2.6-1 of TS 29.222 [13]). | Set to “apiInvokerId”, which uniquely identifies the external MnS consumer (onboarded API invoker).  |
| resOwnerId | The data type of this attribute is defined as "ResOwnerId" and presence qualifier is defined as "O" (see table 8.5.4.2.6-1 of TS 29.222 [13]). |  |
| client\_secret | The data type of this attribute is defined as "string" and presence qualifier is defined as "O" (see table 8.5.4.2.6-1 of TS 29.222 [13]). | If the external MnS consumer has a password, it can be included here.  |
| scope | The data type of this attribute is defined as "string" and presence qualifier is defined as "O" (see table 8.5.4.2.6-1 of TS 29.222 [13]). | It represents the requested scope, i.e. the list of service APIs per AEF that the external MnS consumer requests authorization for. This attribute contains a space-delimitated string as follows: 3gpp#aefId1: apiName, apiName2, … apiNameX; aefId2: apiName1, apiNameapiNameY;…aefIdN:apiName1,apiName2,…apiNameZ  |
| authCode | The data type of this attribute is defined as "string" and presence qualifier is defined as "C" (see table 8.5.4.2.6-1 of TS 29.222 [13]). | RNAA |
| redirect\_uri | The data type of this attribute is defined as "string" and presence qualifier is defined as "O" (see table 8.5.4.2.6-1 of TS 29.222 [13]). | RNAA |

Table 5.1.4.3.x.2-2: Definition of AccessTokenRsp data type

|  |  |  |
| --- | --- | --- |
| Attribute name | Attribute additional information | Comments |
| access\_token | The data type of this attribute is defined as "string" and presence qualifier is defined as "M" (see table 8.5.4.2.7-1 of TS 29.222 [13]). | It represents the access token issued by the CCF.  |
| token\_type | The data type of this attribute is defined as "string" and presence qualifier is defined as "M" (see table 8.5.4.2.7-1 of TS 29.222 [13]). |  |
| expires\_in | The data type of this attribute is defined as "DurationSec" and presence qualifier is defined as "M" (see table 8.5.4.2.7-1 of TS 29.222 [13]). |  |
| scope | The data type of this attribute is defined as "string" and presence qualifier is defined as "O" (see table 8.5.4.2.7-1 of TS 29.222 [13]). | It represents the authorized scope.This attribute is not applicable when access control framework is MSAC.  |

#### 5.1.4.4 Evaluation of potential solutions

#### 5.1.4.4.x Evaluation of potential solution #1

This solution demonstrates how Identity class can be used to define the authorization information for an external MnS consumer. The authentication and authorization MnS producer generates an Identity instance, based on subscription created for the API invoker after service agreement with CAPIF provider. The solution #1 fulfils the requirement PREQ-FS\_MExpo-Auth-0x of the use case.

It is worth noting that this solution is applied offline during the API Invoker enrolment stage, which is a stage not subjected to standardization. Therefore, how steps 1-6 in the solution are executed are at operator’s discretion.

#### 5.1.4.4.a Evaluation of potential solution #x

This solution demonstrates how Identity class can be made available to CCF for an external MnS consumer, so that the CCF can use this information to grant authorization to access one or more service APIs. The authentication and authorization MnS producer sends the MSAC Identity (created during the API invoker enrolment stage) to the CCF. The CCF associates the received MSAC Identity to the “API-invoker-id” (created during the API invoker onboarding procedure).

The solution #x represents a baseline solution; further elaboration on how the authentication and authorization MnS producer generates token claims will be carried out in the normative phase. This baseline solution fulfils the requirement PREQ-FS\_MExpo-Auth-0y of the use case.

|  |
| --- |
| **End Change** |