**3GPP TSG-SA3 Meeting #119 S3-24XXXX**

Orlando, US 11 – 15 November 2024 *revision of S3-244934*

**Source: Nokia**

**Title: Update to KI2 Solution 16 on mapping an API invoker authorization request to the correct CCF in CAPIF interconnect**

**Document for: Approval, Information, Discussion**

**Agenda Item: 5.19**

# 1 Decision/action requested

***Update to Solution on authorization aspect in CAPIF interconnect***

# 2 References

[1] 3GPP TS 23.700-22

[2] 3GPP TS 33.700-22

# 3 Rationale

*Update to Solution on authorization aspect in CAPIF interconnect to address KI2 when authorization code is used in the access token request.*

*The NOTE 1 in Solution details addresses the following EN:*

Editor's Note: Alignment of step 1-5 with existing authorization code flow is ffs.

*The NOTE 2 in Solution details addresses the following EN:*

Editor's Note: Step 5/6 are interrupting the RO.

*As highlighted in the diagram, the communication in step 14 and 15 is between the two CCF of the interconnected domain and the API Invoker initiating the request.*

*The pCR is addressing ENs.*

# 4 Detailed proposal

*\*\*\*\*\*\*\*\*\*\* START OF CHANGES*

## 6.16 Solution #16: Mapping an API invoker authorization request to the correct CCF in CAPIF interconnect

### 6.16.1 Introduction

This solution is addressing KI#2 on security aspects for CAPIF interconnect for APIInvoker authentication and authorization using authorization code flow in CAPIF interconnection (CAPIF-6/6e).

The originator CCF of the API Invoker acts as client towards the interconnected CCF. The originator CCF serves several API invokers and can be connected to several CCFs. When using authorization code the originator CCF gets an authorization code from the CCF of another domain. The authorization code, when received from an API invoker in an access token request, needs to be correlated with the correct CCF. However, the CCF interconnecting with the API invoker’s CCF domain does only provide an authorization code.

The solution proposes how communication between CCF-B (originator CCF) and CCF-A (in the other domain) enables the originator CCF to identify in the access token request of the APIInvoker the correct CCF in the other domain, from which the APIInvoker received an authorization code before.

An example of ROF being in a different domain than API Invoker could be: A user registered with operator 1 is using the friend’s UE which is registered in a different domain (operator 2), to play a game. In this case the API Invoker, since it is installed on friend’s phone, would go to the CCF of operator 2, but to get the game user’s information, the API Invoker should access the information from operator 1, therefore we need the inter-communication between CCF-A and CCF-B

#### 6.16.2.1 Introduction

It is proposed that the originator CCF provides in the authorization code request information about the API invoker and its redirect URI. It receives from the CCF in the other domain the authorization code back and adds a CCF (CCF-A) identifier to the authorization code before sending it to the API invoker. Since the originator CCF needs to handle several API invoker requests, which may be redirected to different CCFs, this allows the originator CCF to send the authorization code together with an access token request from one API invoker to the correct CCF. Hence, the originator CCF is able to handle any subsequent access token request with an authorization code to the correct target CCF in the other domain.

NOTE 1: The following solution assumes that the API invoker and ROF, belongs to two different domains but use the same user-agent during the communication. In the case of API Invoker and ROF not being co-located, a CIBA flow could substitute the authorization code flow, which is however not in scope of this solution.

NOTE 2: Steps 5 and 6 are required only if the API Invoker did not previously granted authorization to the API Invoker.

#### 6.16.2.2 Summary

A CCF receiving from its API invoker a request for authorization code (step 2) adds the API invoker client identifier and its own identifier as it is now acting as client before forwardng the request towards the CCF in the other domain (step 4). Since the other CCF cannot redirect a response to the API invoker, the CCF URI needs to be added too.

Based on the redirect information, the CCF in the other domain provides the authorization code back to the originator CCF (step 7). The originator CCF processes the information before providing the authorization code within its authorization response towards the API invoker. I.e. upon receiving the authorization code, the originator CCF maintains a mapping of API invoker, authorization code and the CCF in the other domain. To map the authorization code (used between the API invoker and the CCF in the other domain, the originator CCF adds (step 8) within the authoriziation response an own identifier for the CCF towards the API invoker (as part of the authorization code). An example is given in the figure below.

In any (subsequent) access token request (step 10) from API invoker using the authorization code, the originator CCF then does the mapping (step11) to the actual identity of the CCF in the other domain and can forward the request via CAPF-6e.

Editor’s Note: Figure update needed.

#### 6.16.2.2 Information flow



Figure 6.162.2-1: Identifying in an API access token request with authorization code the correct CCF in CAPIF interconnect

### 6.16.3 Evaluation

The solution addresses the KI#2 by extending already existing CAPIF authorization solutions to the interconnection scenario.

In particular, the solution enhances the CCF behaviour when receiving an Access Token Request. It introduces new communications between the interconnected CCFs but does not affect the API Invoker or the AEF.

Editor’s note: Further evaluation is ffs.

This solution is specific to RNAA authorization code flow.

*\*\*\*\*\*\*\*\*\*\* END OF CHANGES*