**3GPP TSG-SA3 Meeting #119 S3-245250**

Orlando, US 11 – 15 November 2024 *revision of S3-244937*

**Source: Nokia**

**Title: KI4 Solution on Authentication of the origin API invoker in nested API invocation**

**Document for: Approval**

**Agenda Item: 5.19**

# 1 Decision/action requested

***Addressing missing part in sol 26 on authentication of the origin API invokers in nested API invocation***

# 2 References

[1] 3GPP TS 23.700-22

[2] 3GPP TS 33.700-22

# 3 Rationale

Key issue #4 of TR 33.700-22 addresses nested API invocation. A solution is provided for the authentication of API invoker to the nested API.

# 4 Detailed proposal

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

### 6.X Solution: Authentication of the origin API invoker in nested API invocation

### 6.X.1 Introduction

The solution addresses the authentication part of KI#4 on nested API invocation.

Before an API invoker can request the access token to invoke a service API exposed by AEF 1 it needs to get authenticated. However, if AEF1 decides to invoke another service API exposed by AEF2 that relates to the origin API Invoker, the token issued by CCF and finally received by AEF 2 does not authenticate the API invoker against AEF2.

The solution proposes to use the CCA concept. The origin API invoker includes CCA into its service request to AEF1, such that AEF1, if it needs to invoke another service from AEF2, can request an access token on behalf of the origin API invoker by presenting CCA to CCF.

### 6.X.2 Solution Details

1) API invoker and CCF have mutual authenticated.

2) API invoker discovers at CCF the AEF1.

3) CCF indicates in its service discovery response that nested API invocation is supported at AEF1. The indication allows the API invoker to include CCA for using the service later on. Eg. use of CCA required for nested API support.

4) The API invoker requests an Access Token for AEF1.

5) CCF returns the access token.

6) API invoker mutually authenticates with the discovered AEF 1 and hence, establishes a secure connection.

7) If nested API support was indicated during discovery, the API invoker generates a client credential assertion token (CCA) including the API Invoker Id, the CCF address and other information such as the expiry time of the CCA.

8) The Service request including the CCA is then sent to the AEF 1.

9) AEF1 receives the Service request including CCA. If AEF1 needs to invoke another service, it acts as an API invoker to AEF2.

10) AEF1 has established mTLS with CCF and sends its own AccessTokenRequest to CCF to access the service of AEF2 on behalf of the origin API invoker. This includes CCA, optionally it includes also the previously received token.

11) CCF will authenticate the origin APIInvoker via CCA and authorizes APIInvoker and AEF1 by provisioning a new access token to be used by AEF1 to access the nested service from AEF2 on behalf of the origin API invoker.

12) CCF provides the new access token to AEF1.

13) AEF1 creates the Service request to AEF2 along with the CCA received from the origin API invoker and the access token received in step 12.

14) AEF2 authenticate the origin API invoker via CCA and validates that the API invoker Id in CCA is matched with API invoker Id in the access token.



### 6.X.3 Evaluation

The solution assumes that CCF knows in advance that the AEF1 will invoke AEF2 API.

The solution requires a client credential authentication token to be introduced to allow for authentication when the nested API invocation scenario needs to authenticate the API Invoker.

The following entities are impacted:

API Invoker: The solution requires that the API invoker includes a CCA token to allow the final AEF to authenticate the API invoker in a nested API invocation scenario.

CCF/AEF: The solution requires CCF to process an access token request twice and AEF1 to request an access token and to update the service request (CCF authenticates and authorizes the API invoker of the original request.Then CCF needs to authenticate the access token request also from AEF1 to allow AEF1 to request the service with the updated token for service invocation from AEF2).

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