**3GPP TSG-SA5 Meeting #141-e *S5-222xxx***

**e-meeting, 17 - 25 January 2022**

|  |
| --- |
| *CR-Form-v12.1* |
| **CHANGE REQUEST** |
|  |
|  | **32.158** | **CR** | **xx** | **rev** | - | **Current version:** |  |  |
|  |
| *For* [***HE******LP***](http://www.3gpp.org/3G_Specs/CRs.htm#_blank)*on using this form: comprehensive instructions can be found at* [*http://www.3gpp.org/Change-Requests*](http://www.3gpp.org/Change-Requests)*.* |
|  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Proposed change affects:*** | UICC apps |  | ME |  | Radio Access Network | **X** | Core Network | **X** |

|  |
| --- |
|  |
| ***Title:***  | Enhance OpenAPI to support access control  |
|  |  |
| ***Source to WG:*** | Nokia, Nokia Shanghai Bell |
| ***Source to TSG:*** | S5 |
|  |  |
| ***Work item code:*** | MSAC |  | ***Date:*** | 2022-01-07 |
|  |  |  |  |  |
| ***Category:*** | B |  | ***Release:*** | Rel-17 |
|  | *Use one of the following categories:****F*** *(correction)****A*** *(mirror corresponding to a change in an earlier release)****B*** *(addition of feature),* ***C*** *(functional modification of feature)****D*** *(editorial modification)*Detailed explanations of the above categories canbe found in 3GPP [TR 21.900](http://www.3gpp.org/ftp/Specs/html-info/21900.htm). | *Use one of the following releases:Rel-8 (Release 8)Rel-9 (Release 9)Rel-10 (Release 10)Rel-11 (Release 11)…Rel-15 (Release 15)Rel-16 (Release 16)Rel-17 (Release 17)Rel-18 (Release 18)* |
|  |  |
| ***Reason for change:*** | According to conclusion of TR 28.817 and discussion paper agreed in last meeting, the generic management service of OpenAPI should be updated to support authentication and authorization. |
|  |  |
| ***Summary of change:*** | Enhance OpenAPI to support authentication and authorization capability |
|  |  |
| ***Consequences if not approved:*** | No standardized way to support acess control for OpenAPI. |
|  |  |
| ***Clauses affected:*** | 2, 6.x (new), Annex X(new), Annex Y(new) |
|  |  |
|  | **Y** | **N** |  |  |
| ***Other specs*** |  | **X** |  Other core specifications  | TS/TR ... CR ...  |
| ***affected:*** |  | **X** |  Test specifications | TS/TR ... CR ...  |
| ***(show related CRs)*** |  | **X** |  O&M Specifications | TS/TR ... CR ...  |
|  |  |
| ***Other comments:*** | Resubmission of S5-221365 |
|  |  |
| ***This CR's revision history:*** |  |

|  |
| --- |
| **Start of 1st modification** |

# 2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non‑specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] IETF RFC 7231: "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content".

[3] 3GPP TS 32.300: "Telecommunication management; Configuration Management (CM); Name convention for Managed Objects".

[4] IETF RFC 3986: "Uniform Resource Identifier (URI): Generic Syntax".

[5] IETF RFC 7230: "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing".

[6] IETF RFC 7159: " The JavaScript Object Notation (JSON) Data Interchange Format".

[7] draft-wright-json-schema-01 (October 2017): "JSON Schema: A Media Type for Describing JSON Documents".

 Editor's note: The above document cannot be formally referenced until it is published as an RFC.

[8] draft-wright-json-schema-validation-01 (October 2017: "JSON Schema Validation: A Vocabulary for Structural Validation of JSON".

 Editor's note: The above document cannot be formally referenced until it is published as an RFC.

[9] draft-wright-json-schema-hyperschema-01 (October 2017): "JSON Hyper-Schema: A Vocabulary for Hypermedia Annotation of JSON.

 Editor's note: The above document cannot be formally referenced until it is published as an RFC.

[10] OpenAPI Specification (<https://github.com/OAI/OpenAPI-Specification>)

[11] IETF RFC 5789: "PATCH Method for HTTP".

[12] IETF RFC 7396: "JSON Merge Patch".

[13] IETF RFC 6902: "JavaScript Object Notation (JSON) Patch".

[14] IETF RFC 6901: "JavaScript Object Notation (JSON) Pointer".

[15] XML Path Language (XPath) Version 1.0, W3C Recommendation 16 November 1999 (<https://www.w3.org/TR/xpath-10/>)

[16] 3GPP TR 32.160: "Management and orchestration; Management service template".

[x] IETF RFC 6749: "The OAuth 2.0 Authorization Framework".

[y] OpenID: OpenID connect protocol:

 <https://openid.net/specs/openid-connect-core-1_0.html>

[z] IETF RFC 7519: "JSON Web Token (JWT)".

|  |
| --- |
| **End of modification** |

|  |
| --- |
| **Start of 2nd modification** |

## 6.x Access control service

### 6.x.1 RESTful HTTP-based solution set

#### 6.x.1 Definition of access control services

##### 6.x.1.1 Introduction

The OpenID connect protocol (see [y]) and OAuth 2.0 Authorization Code Grant (see RFC 6749 [x]) are used to authenticate and authorize human management service consumer, as shown in Figure 6.x.1.1-1.



**Figure 6.x.1.1-1 Authentication and authorization for human MnS consumer**

* The authentication service producer implements OpenID Provider (OP), and authorization endpoint of OAuth 2.0 which authenticates the end user in OAuth 2.0 authorization code grant scenario.
* The MnS consumer is end user of OpenID connect protocol and resource owner of OAuth 2.0.
* The client acting on behalf of human MnS consumer implements Relaying Party (RP) of OpenID connect protocol, and confidential client of OAuth 2.0.
* The authorization service producer implements token endpoint of OAuth 2.0, which issue access token to the client.
* The management service producer implements resource server of OAuth 2.0.

The procedure is as follows:

101) The RP (Client on behalf of MnS consumer) sends an authentication request to the OpenID Provider (OP).

102) The OP authenticates the MnS consumer and authentication assertion.

103) Authentization service producer returns authentication assertion to the MnS consumer once successfully authenticate the MnS consumer.

104) With the authentication assertion, the MnS consumer asks permissions from authentication service producer via Authorization Request.

105) Authorization service producer validates the assertion, check the role/group of the MnS consumer and assigns related permissions to the MnS consumer. Authorization service producer constructs access token based on granted permissions and sends back the token to the MnS consumer.

106) The MnS consumer checks the access token, construct a service request based on permissions in the token, and sends service request, including access token, to MnS producer. MnS producer validates the access token, perform the request and return result to the MnS consumer

Note: Authentication of human MnS consumer includes two steps, the client on behalf of human user sends authentication request is the first step, and the human user logins with credentials is the second step. The steps to redirect user agent (e.g. browser) to authentication service producer (by the client on behalf of MnS consumer) to send authentication request, and redirect user agent back to the client (by authentication service producer) to send authentication response are ignored in the figure for sake of simplicity and conciseness.

OAuth 2.0 Client Credentials Grant (see RFC 6749 [x]) is used to authenticate and authorize machine management service consumer, as shown in Figure 6.x.1.1-2.



**Figure 6.x.1.1-2 Authentication and authorization for machine MnS consumer**

* The authentication service producer authenticates management service consumer by validate the client credential.
* The management service consumer implements confidential client of OAuth 2.0.
* The authorization service producer implements token endpoint of OAuth 2.0, which issue access token to the client.
* The management service producer implements resource server of OAuth 2.0.

The procedure is as follows:

101) The Management Service Consumer authenticates with the authorization server and requests an access token

102,103) The authentication service producer authenticates management service consumer by validate the client credential

104) After the authorization server authenticating the client; the authorization service producer issues an access token.

105) The MnS consumer checks the access token, construct a service request based on permissions in the token, and sends service request, including access token, to MnS producer. MnS producer validates the access token, perform the request and return result to the MnS consumer

Note: Authentication service producer is preconfigured in authorization service producer, and vice versa. Trust relationship between the two entities is established.

The IS operations are mapped to SS equivalents according to table 6.x.1.1-1.

Table 6.x.1.1-1: Implement access control services in OpenAPI SS

|  |  |  |  |
| --- | --- | --- | --- |
| Access control service | HTTP Method | Resource URI | S |
| authentication | GET | /oauth2/authorize | M |
| authorization | POST | /oauth2/token | M |

|  |
| --- |
| **End of modification** |

|  |
| --- |
| **Start of 3rd modification** |

##### Annex X (Normative):

# X.1 Authentication

Map NRM to OpenAPI parameters according to table X.1-1 and table X.1-2.

Table X.1-1: Mapping NRM to OpenAPI input parameters (HTTP GET)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SS parameter location | SS parameter name | NRM attribute | S | Remark |
| query | consumer\_id | Identity4AC.identifier | M | A unique identifier of a MnS consumer. For machine MnS consumer, it could be DN, FQDN, etc. It is included in authentication request.For human MnS consumer, it could be user name, email address, phone number, etc. It is included in both of authentication request from the client (acting on behalf of human MnS consumer) to authentication service producer and login request from user agent to authentication service producer. |
| query | credential\_type | Identity4AC.credentialType | CM | Different credential types will be used according to authentication policy of the MnS consumer, e.g. it could be secret (e.g. password) or certificate based assertion (e.g. jwt-bear, see RFC 7519 [z])For human MnS consumer, it is included in login request from user agent to authentication service producer.For machine MnS consumer, it is in authentication request. |
| query | credential | Identity4AC.credential | CM | It is secret or certificate based assertion.For human MnS consumer, it is included in login request from user agent to authentication service producer.For machine MnS consumer, it is in authentication request. |
| query | client\_id | Identity4AC.authSession.assocClient  | CM | It is used only for human MnS consumer scenario. It is part of associated client acting on behalf of the human consumer. It is an unique id, e.g. DN, FQDN, assigned to the client. |
| query | redirect\_uri | Identity4AC.authSession.assocClient  | CM | It is used only for human MnS consumer scenario. It is part of associated client acting on behalf of the human consumer. It is redirection URI to which the authentication response from authentication service producer will be sent. |
| query | response\_type | NA | CM | It is oauth2 and OpenID connect specific parameter.It presents and its value is "code" in the authentication request from the client (acting on behalf of human MnS consumer) to authentication service producer for human MnS consumer authentication. It is empty in the authentication request from authorization service producer (acting on behalf of machine MnS consumer) to authentication service producer for machine MnS consumer authentication |
|  |  |  |  |  |

Table X.1-2: Mapping NRM to OpenAPI output parameters (HTTP GET))

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SS parameter location | SS parameter name | NRM attribute | S | Remark |
| response status codes/body | status | NA | M | It is response status code, and optional error description in response body for error response. |
| response body | consumer\_id | Identity4AC.identifier | M | same to identifier in the request |
| response body | session\_id | Identity4AC.authSession.sessionId | O | It could be used to uniquely identify a successful authentication for a MnS consumer in different entities included in authentication workflow. |
| response body | code | Identity4AC.authSession.assertion | CM | It is oauth2 and OpenID connect specific parameter.It presents and its value is set to authorization code generated by the authentication service producer. It's only applicable to human management service consumer. |
|  |  |  |  |  |

# X.2 Authorization

Map NRM to OpenAPI parameters according to table X.2-1 and table X.2-2.

Table X.2-1: Mapping NRM to OpenAPI input parameters (HTTP POST)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SS parameter location | SS parameter name | NRM attribute | S | Remark |
| query | grant\_type | NA | M | It is oauth2 specific parameter used to designate how to authenticate a client. It's set to "authorization\_code" for authorization of human management service consumer, and "client\_credentials" for authorization of machine management service consumer |
| query | code | Identity4AC.authSession.assertion | CM | It is oauth2 and OpenID connect specific parameter.It presents and its value is set to authorization code generated by the authentication service producer. It's only applicable to human management service consumer. |
| query | client\_id | Identity4AC.authSession.assocClient  | CM | It is used only for human MnS consumer scenario. It is part of associated client acting on behalf of the human consumer. It is unique id, e.g. DN, FQDN, assigned to the client. |
| query | redirect\_uri | Identity4AC.authSession.assocClient  | CM | It is used only for human MnS consumer scenario. It is part of associated client acting on behalf of the human consumer. It is redirection URI to which the authentication response from authentication service producer will be sent. |
| query | scope | AccessRight | O | The scope of access request |

Table X.2-2: Mapping NRM to OpenAPI output parameters (HTTP POST)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SS parameter location | SS parameter name | NRM attribute | S | Remark |
| response status codes/body | status | NA | M | It is response status code, and optional error description in response body for error response. |
| response body | access\_token | Identity4AC.authSession.accessToken | CM | It is the access token issued by the authorization server |
| response body | token\_type | Identity4AC.authSession.accessToken.tokenType | CM | It is type of the access token. |

**Then access\_token will be put into http "authorization" header when access management services with access control. e.g. Authorization: Basic SlAV32hkKG**

##### Annex Y (informative):

# Y.1 Informative example: human MnS consumer authentication and authorization :

Authentication request from clien on behalf of human MnS consumer to authentication service producer:

 GET /oauth2/authorize?

consumer\_id=consumer1@example.com

 &client\_id=client.example.com

 &redirect\_uri=https%3A%2F%2Fclient.example.com%2Fac

 &response\_type=code

 Host: authenticationserver.example.com

Login request from user agent to authentication service producer:

 GET /oauth2/authorize?

consumer\_id=consumer1@example.com

&credential\_type=secret

&credential=SHJKUJUYKKLH

 Host: authenticationserver.example.com

Authentication response:

 HTTP/2 302 Found

 Location: https://client.example.org/ac?

consumer\_id=consumer1@example.com

&code=SplxlOBeZQQYbYS6WxSbIA

Authorization request with granted code

POST /oauth2/token?

grant\_type=authorization\_code

&code=SplxlOBeZQQYbYS6WxSbIA

 &client\_id=client.example.com

 &redirect\_uri=https%3A%2F%2Fclient.example.com%2Fac

Host: authorizationserver.example.com

Authorization response

 HTTP/1.1 200 OK

 Content-Type: application/json

 Cache-Control: no-store

 Pragma: no-cache

 {

 "access\_token": "SlAV32hkKG",

 "token\_type": "Bearer",

 "context": "expire in 60m"

 }

# Y.2 Informative example: machine MnS consumer authentication and authorization:

Authentication and authorization request:

 POST /oauth2/token?

grant\_type=client\_credentials

&consumer\_id=consumer1.example.com

&credential\_type=jwt

&credential=eyJhbGciOiJSUzI1NiIsIng1dCI6Imd4OHRHeXN5amNScUtq

 Host: authorizationserver.example.com

Authentication and authorization response:

 HTTP/1.1 200 OK

 Content-Type: application/json

 Cache-Control: no-store

 Pragma: no-cache

 {

 "access\_token": "SlAV32hkKG",

 "token\_type": "Bearer",

 "context": "expire in 60m"

 }

|  |
| --- |
| **End of modification** |