**3GPP TSG-SA5 Meeting #144-e *S5-224071***

**e-meeting,, 27 June - 1 July 2022**

|  |
| --- |
| *CR-Form-v12.1* |
| **CHANGE REQUEST** |
|  |
|  | **28.532** | **CR** | **DraftCR** | **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:***  | Rel-18 TS28.532 enhance OpenAPI to support access control |
|  |  |
| ***Source to WG:*** | Nokia, Nokia Shanghai Bell |
| ***Source to TSG:*** | S5 |
|  |  |
| ***Work item code:*** | MSAC |  | ***Date:*** | 2022-06-17 |
|  |  |  |  |  |
| ***Category:*** | B |  | ***Release:*** | Rel-18 |
|  | *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 requirements related to access control, 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, 12.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:*** | This is revision of S5-223399.This is input to the Rel-18 28.532 DraftCR for . |
|  |  |
| ***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] 3GPP TS 28.526: "Telecommunication management; Life Cycle Management (LCM) for mobile networks that include virtualized network functions; Procedures".

[3] 3GPP TS 28.541: "Management and orchestration ; 5G Network Resource Model (NRM); Stage 2 and stage3".

[4] ITU-T Recommendation X.733 (02/92): "Information technology - Open Systems Interconnection - Systems Management: Alarm reporting function".

[5] 3GPP TS 28.531: "Management and orchestration ; Provisioning; ".

[6] 3GPP TS 28.554: "Management and orchestration ; 5G end to end Key Performance Indicators (KPI)".

[7] 3GPP TS 22.261: "Technical Specification Group Services and System Aspects; Service requirements for the 5G system; Stage 1".

[8] 3GPP TS 23.501: "Technical Specification Group Services and System Aspects; System Architecture for the 5G System; Stage 2".

[9] 3GPP TS 23.003: "Technical Specification Group Core Network and Terminals; Numbering, addressing and identification".

[10] ETSI GS NFV-IFA 013 V2.4.1 (2018-02) "Network Function Virtualization (NFV); Management and Orchestration; Os-Ma-nfvo Reference Point - Interface and Information Model Specification".

[11] 3GPP TS 28.622: "Telecommunication management; Generic Network Resource Model (NRM) Integration Reference Point (IRP); Information Service (IS)".

[12] ETSI GS NFV-IFA 015 (V2.4.1): "Network Function Virtualisation (NFV); Management and Orchestration; Report on NFV Information Model".

[13] 3GPP TS 28.533: "Management and orchestration; Architecture framework"

[14] ITU-T Recommendation X.734 (1992): "Information technology - Open Systems Interconnection - Systems management: Event report management function".

[15] 3GPP TS 32.158: "Management and orchestration; Design rules for REpresentational State Transfer (REST) Solution Sets (SS)".

[16] 3GPP TS 32.302: "Telecommunication management; Configuration Management (CM); Notification Integration Reference Point (IRP); Information Service (IS)".

[17] 3GPP TS 32.401: "Telecommunication management; Performance Management (PM); Concept and requirements".

[18] 3GPP TS 28.552: "Management and orchestration; 5G performance measurements".

[19] 3GPP TS 32.401: "Telecommunication management; Perfomance Measurement (PM); Concept and requirements".

[20] ISO 8601:2004: "Data elements and interchange formats – Information interchange – Representation of dates and times".

[21] Void.

[22] Void.

[23] Void.

[24] Void.

[25] 3GPP TS 32.300: "Telecommunication management; Configuration Management (CM); Name convention for Managed Objects ".

[26] W3C REC-xmlschema-0-20010502: "XML Schema Part 0: Primer".

[27] W3C REC-xmlschema-1-20010502: "XML Schema Part 1: Structures".

[28] W3C REC-xmlschema-2-20010502: "XML Schema Part 2: Datatypes".

[29] W3C REC-xml-names-19990114: "Namespaces in XML".

[30] Void.

[31] 3GPP TS 32.111-2: " Telecommunication management; Fault Management; Part 2: Alarm Integration Reference Point (IRP): Information Service (IS)".

[32] IETF RFC 6241 "Network Configuration Protocol (NETCONF)".

[33] 3GPP TS 32.160 " Management and orchestration; Management service template ".

[34] IETF RFC 7950 "The YANG 1.1 Data Modeling Language".

[35] OpenAPI: "OpenAPI 3.0.1 Specification", <https://github.com/OAI/OpenAPI-Specification/blob/master/versions/3.0.1.md>.

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

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

[38] 3GPP TS 32.422: "Telecommunication management; Subscriber and equipment trace; Trace control and configuration management".

[39] 3GPP TS 32.423: "Telecommunication management; Subscriber and equipment trace; Trace data definition and management".

[40] IETF RFC 6455: "The WebSocket Protocol".

[41] IETF RFC 793: "Transmission Control Protocol".

[42] 3GPP TS 28.550: "Management and orchestration; Performance assurance".

[43] ITU-T Recommendation X.733 (02/92): "Information technology - Open Systems Interconnection - Systems Management: Alarm reporting function".

[44] 3GPP TS 28.623: "Telecommunication management; Generic Network Resource Model (NRM) Integration Reference Point (IRP); Solution Set (SS) definitions".

[45] Text Attribution: Creator: ONAP, under Creative Commons Attribution 4.0 International License, https://creativecommons.org/licenses/by/4.0/, URI to access the text: <https://github.com/onap/vnfrqts-requirements/blob/05f26fac2b941513a7d0e856b99fd8c61d688299/docs/Chapter8/ves7_1spec.rst#resource-structure>.

[46] 3GPP SA5 FORGE OpenAPI definitions: <https://forge.3gpp.org/rep/sa5/MnS/tree/Rel-16/OpenAPI>.

[47] 3GPP TS 32.404: "Performance Management (PM); Performance measurements; Definitions and template".

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

[49] IETF RFC 8040: "RESTCONF protocol".

[50] IETF RFC 7951: " JSON Encoding of Data Modeled with YANG".

[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)".

[a] IETF RFC 7515: " JSON Web Signature (JWS)"

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

|  |
| --- |
| **Start of next modification** |

## 12.x Access control service

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

#### 12.x.1.1 Definition of access control services

##### 12.x.1.1.1 Introduction

Solution shall be based on OpenID connect protocol (see [y]) and OAuth 2.0 (see RFC 6749 [x]).

OpenID Connect is a simple identity layer on top of the OAuth 2.0 protocol and used in authentication. It enables clients to verify the identity of the MnS Consumer based on the authentication performed by an Authorization Service Producer.

OAuth introduces an authorization layer and separates the role of the client (MnS Consumer) from that of the resource owner (e.g., Operator). In OAuth, the client requests access to resources controlled by the resource owner and hosted by the resource server (MnS Producer), and is issued a different set of credentials than those of the resource owner.

Instead of using the resource owner's credentials to access protected resources, the client obtains an access token -- a string denoting a specific scope, lifetime, and other access attributes. Access tokens are issued to clients by an authorization service producer with the approval of the resource owner. The client uses the access token to access the protected resources hosted by the resource server.

In OAuth, an authorization grant is a credential representing the resource owner's authorization (to access its protected resources) used by the client to obtain an access token. To request an access token, the client obtains authorization from the resource owner. The authorization is expressed in the form of an authorization grant, which the client uses to request the access token.

OAuth2.0 specifies four grant types in clause 1.3 (see RFC 6749 [x]). The access control shall support the following two types of grants:

(1) authorization code

(2) client credential

All other procedure related to these two grant types is used.

In OAuth the access taken is a string. The access policies of this string are not defined by OAuth. This solution adds on top by defining how to specify these access policies. In OAuth authorization request, the scope parameter is optional. With OpenID connect, the scope parameter is required to have value "openid", other values are optional. In Access control, the scope parameter is not used for access right scope during authentication / authorization request. The access right is provisioned for the identities during preparation phase or runtime phase, refer to TS28.533 [13].

##### 12.x.1.1.2 human management service consumer authentication and authorization

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 12.x.1.1.2-1. To be noted, the workflows in Figure 12.x.1.1.2-1 and 12.x.1.1.3-1 are runtime workflows, refer to TS 28.533 [13] for the preconditions related to access rights provisioning.



**Figure 12.x.1.1.2-1 Authentication and authorization for human MnS consumer**

In OpenID connect protocol, OAuth 2.0 Server implementing OpenID Connect protocol are also referred to as OpenID Providers (OPs). OAuth 2.0 Clients using OpenID Connect are also referred to as Relying Parties (RPs).

In this solution, the authentication service producer takes role of 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 via user agent is end user of OpenID connect protocol.

The client acting on behalf of human MnS consumer takes the role of Relaying Party (RP) of OpenID connect protocol, and confidential client of OAuth 2.0.

An OpenID Connect Authentication Request is an OAuth 2.0 Authorization Request that requests that the End-User to be authenticated by the Authorization Service Producer.

The authorization service producer takes role of token endpoint of OAuth 2.0, which issues access token to the client.

The MnS producer plays the role of the resource server.

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.

##### 12.x.1.1.3 machine management service consumer authentication and authorization

OAuth 2.0 client credential grant (see RFC 6749 [x]) is used to authenticate and authorize machine management service consumer, as shown in Figure 12.x.1.1.3-1.



**Figure 12.x.1.1.3-1 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.

Note: Authentication service producer is preconfigured in authorization service producer (e.g., certification configuration of both sides, access information of both sides such as name, or address, etc. ), and vice versa. Trust relationship between the two entities is established.

The access control service is implemented in OpenAPI according to table 12.x.1.1.3-1.

Table 12.x.1.1.3-1: 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 |

#### 12.x.1.2 authentication

Map NRM to OpenAPI parameters according to table 12.x.1.2-1 and table 12.x.1.2-2.

Table 12.x.1.2-1: Mapping NRM to OpenAPI input parameters (HTTP GET)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NRM IS attribute name | NRM SS attribute name | SS parameter location | S | Remark |
| Identity4AC.identifier | consumer\_id | query | 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 MnS consumer via human user, it could be user name, email address, phone number, etc. It is included in both of authentication request from the client (actioning on behalf of human user) to authentication service producer and login request from user agent to authentication service producer.The parameter consumer\_id is introduced in access control solution in addition to parameters defined in OAuth2.0. |
| Identity4AC.credentialType | credential\_type | query | 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.The parameter credential\_type is introduced in access control solution in addition to parameters defined in OAuth2.0. |
| Identity4AC.credential | credential | query | 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.The parameter credential is introduced in access control solution in addition to parameters defined in OAuth2.0. |
| Identity4AC.authSession.assocClient  | client\_id | query | 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.The parameter client\_id is defined in OAuth2.0. |
| Identity4AC.authSession.assocClientUri  | redirect\_uri | query | 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.The parameter redirect\_uri is defined in OAuth2.0. |
|  | response\_type | query | CM | It is oauth2 and OpenID connect specific parameter.It presents and its value is "code" in the authentication request from the client (actioning 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 (actioning on behalf of machine MnS consumer) to authentication service producer for machine MnS consumer authenticationThe parameter response\_type is defined in OAuth2.0. |
|  | scope | query | CM | It is used only for MnS consumer via human user scenario. OpenID Connect requests shall contain the "openid" as scope value.The parameter scope is defined in OAuth2.0. |
|  |  |  |  |  |

Table 12.x.1.2-2: Mapping NRM to OpenAPI output parameters (HTTP GET))

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SS parameter location | SS parameter name | NRM attribute | S | Remark |
| response status codes/body | status |  | 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.The parameter consumer\_id is introduced in access control solution in addition to parameters defined in OAuth2.0. |
| response body | session\_id | Identity4AC.authSession.sessionId | O | It could be used to uniquely identity a successful authentication for a MnS consumer in different entities included in authentication workflow.The parameter session\_id is introduced in this solution in addition to parameters defined in OAuth2.0. |
| 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.The parameter code is defined in OAuth2.0. |
|  |  |  |  |  |

#### 12.x.1.3 authorization

Map NRM to OpenAPI parameters according to table 12.x.1.3-1 and table 12.x.1.3-2.

Table 12.x.1.3-1: Mapping NRM to OpenAPI input parameters (HTTP POST)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SS parameter location | SS parameter name | NRM attribute | S | Remark |
| query | grant\_type |  | 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\_credential " for authorization of machine management service consumerThe parameter grant\_type is defined in OAuth2.0. |
| 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.The parameter code is defined in OAuth2.0. |
| 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.The parameter client\_id is defined in OAuth2.0. |
| 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.The parameter redirect\_uri is defined in OAuth2.0. |
|  |  |  |  |  |

Table 12.x.1.3-2: Mapping NRM to OpenAPI output parameters

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SS parameter location | SS parameter name | NRM attribute | S | Remark |
| response status codes/body | status |  | 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.The access token shall be a JSON Web Token (JWT) as specified in IETF RFC 7519 [z]. The access token shall include the claims encoded as a JSON object and then digitally signed using JWS as specified in IETF RFC 7515 [a] and in clause 13.4.1 of 3GPP TS 33.501 [8].The digitally signed access token shall be converted to the JWS Compact Serialization encoding as a string as specified in clause 7.1 of IETF RFC 7515 [a].The parameter access\_token is defined in OAuth2.0. |
| response body | token\_type | Identity4AC.authSession.accessToken.tokenType | CM | It is type of the access token.The parameter token\_type is defined in OAuth2.0. |

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

|  |
| --- |
| **Start of next modification** |

Annex Y (Informative):
Informative example for authentication and authorization

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

**Informative example: human MnS consumer authentication and authorization :**

Authentication request sent from a client on behalf of a human MnS consumer to an 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\

 &scope=openid

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:

**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** |