|  |  |
| --- | --- |
| 3GPP TR 33.721 V0.7.0 (2025-02) | |
| Technical Report | |
| 3rd Generation Partnership Project;  Technical Specification Group Services and System Aspects;  Study on security aspects of 5G Mobile Metaverse services;  (Release 19) | |
|  | |
|  | 3GPP-logo_web |
| The present document has been developed within the 3rd Generation Partnership Project (3GPP TM) and may be further elaborated for the purposes of 3GPP. The present document has not been subject to any approval process by the 3GPPOrganizational Partners and shall not be implemented. This Specification is provided for future development work within 3GPPonly. The Organizational Partners accept no liability for any use of this Specification. Specifications and Reports for implementation of the 3GPP TM system should be obtained via the 3GPP Organizational Partners' Publications Offices. | |

|  |
| --- |
|  |
| ***3GPP***  Postal address  3GPP support office address  650 Route des Lucioles - Sophia Antipolis  Valbonne - FRANCE  Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16  Internet  http://www.3gpp.org |
| ***Copyright Notification***  No part may be reproduced except as authorized by written permission. The copyright and the foregoing restriction extend to reproduction in all media.  © 2024, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).  All rights reserved.  UMTS™ is a Trade Mark of ETSI registered for the benefit of its members  3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners LTE™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners  GSM® and the GSM logo are registered and owned by the GSM Association |

Contents

Foreword 5

Introduction 6

1 Scope 7

2 References 7

3 Definitions of terms, symbols and abbreviations 8

3.1 Terms 8

3.2 Symbols 8

3.3 Abbreviations 8

4 Security assumptions 8

5 Key issues 8

5.1 Key Issue #1: Authorization supporting spatial localization service 9

5.1.1 Key issue details 9

5.1.2 Security threats 9

5.1.3 Potential security requirements 9

5.2 Key Issue #2: Privacy of user sensitive information 9

5.2.1 Key issue details 9

5.2.2 Security threats 9

5.2.3 Potential security requirements 10

5.3 Key issue #3: Security aspects of digital asset container in 5G 10

5.3.1 Key issue details 10

5.3.2 Security threats 10

5.3.3 Potential security requirements 10

5.4 Key Issue #4: Authentication and authorization of digital representation 10

5.4.1 Key issue details 10

5.4.2 Security threats 11

5.4.3 Potential security requirements 11

5.X Key Issue #X: <Key Issue Name> 11

5.X.1 Key issue details 11

5.X.2 Security threats 11

5.X.3 Potential security requirements 11

6 Solutions 11

6.1 Solution #1: Support for spatial localization service authorization 12

6.1.1 Introduction 12

6.1.2 Solution details 12

6.1.3 Evaluation 13

6.2 Solution #2: Solution for KI#1 on Authorization supporting spatial localization service with CAPIF Core Function (CCF) 14

6.2.1 Introduction 14

6.2.2 Solution details 14

6.2.2.1 Procedure of authorization for spatial localization service 14

6.2.3 Evaluation 15

6.3 Solution #3: Solution for KI#1 on Authorization supporting spatial localization service with CAPIF Core Function (CCF) in case of different resource owner 15

6.3.1 Introduction 15

6.3.2 Solution details 16

6.3.2.1 Procedure of authorization of spatial anchor service with multiple VAL server 16

6.3.3 Evaluation 17

6.4 Solution #4: Privacy protection for user sensitive information exposure 17

6.4.1 Introduction 17

6.4.2 Solution details 17

6.4.3 Evaluation 18

6.5 Solution #5: Privacy protection during metaverse service discovery 18

6.5.1 Introduction 18

6.5.2 Solution details 18

6.5.3 Evaluation 20

6.6 Solution #6: Digital asset request validation 20

6.6.1 Introduction 20

6.6.2.1 Access token request/response 20

6.6.2.2 Solution details 21

6.7 Solution #7: Authorize avatar by metaverse service provider 22

6.7.1 Introduction 22

6.7.2 Solution details 22

6.7.3 Evaluation 23

6.8 Solution #8: Authenticate and authorize DA client to create a digital asset 23

6.8.1 Introduction 23

6.8.2 Solution details 24

6.8.3 Evaluation 26

6.9 Solution #9: Authenticate and authorize VAL server/client to access digital asset 26

6.9.1 Introduction 26

6.9.2 Solution details 26

6.9.3 Evaluation 28

6.10 Solution #10: Avatar authentication and authorization supported by A-DACM architecture 28

6.10.1 Introduction 28

6.10.2 Solution details 29

6.10.2.1 Description 29

6.10.2.2 Avatar authentication procedure 29

6.10.2.3 Avatar authorization procedure 31

6.10.3 Evaluation 32

6.11 Solution #11: Avatar authentication and authorization supported by CAPIF 32

6.11.1 Introduction 32

6.11.2 Solution details 32

6.11.3 Evaluation 33

6.Y Solution #Y: <Solution Name> 33

6.Y.1 Introduction 33

6.Y.2 Solution details 34

6.Y.3 Evaluation 34

7 Conclusions 34

7.1 Conclusion on Key Issue #1 34

7.2 Conclusion on Key Issue #2 34

7.3 Conclusion on Key Issue #3 34

7.4 Conclusion on Key Issue #4 35

Annex <X> (informative): Change history 36

# Foreword

This Technical Report has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

In the present document, modal verbs have the following meanings:

**shall** indicates a mandatory requirement to do something

**shall not** indicates an interdiction (prohibition) to do something

The constructions "shall" and "shall not" are confined to the context of normative provisions, and do not appear in Technical Reports.

The constructions "must" and "must not" are not used as substitutes for "shall" and "shall not". Their use is avoided insofar as possible, and they are not used in a normative context except in a direct citation from an external, referenced, non-3GPP document, or so as to maintain continuity of style when extending or modifying the provisions of such a referenced document.

**should** indicates a recommendation to do something

**should not** indicates a recommendation not to do something

**may** indicates permission to do something

**need not** indicates permission not to do something

The construction "may not" is ambiguous and is not used in normative elements. The unambiguous constructions "might not" or "shall not" are used instead, depending upon the meaning intended.

**can** indicates that something is possible

**cannot** indicates that something is impossible

The constructions "can" and "cannot" are not substitutes for "may" and "need not".

**will** indicates that something is certain or expected to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

**will not** indicates that something is certain or expected not to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

**might** indicates a likelihood that something will happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

**might not** indicates a likelihood that something will not happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

In addition:

**is** (or any other verb in the indicative mood) indicates a statement of fact

**is not** (or any other negative verb in the indicative mood) indicates a statement of fact

The constructions "is" and "is not" do not indicate requirements.

# Introduction

Editor's Note: The introduction clause content is left for future consideration.

# 1 Scope

The present document studies security impacts of the procedures introduced in Study on Application enablement architecture for mobile metaverse services studied in TR 23.700-21[2], specifically, the security aspects that are to be covered in this study are as follows:

- authentication and authorization of digital identity (non-IMS based)

NOTE: The term digital identity is defined in clause 3.1.

- support security aspects of digital asset container

 Editor's Note: Whether the digital asset container is specified in 5GC or in the application layer is under the remit of SA6.

- security and privacy aspects of user sensitive information for Localized Mobile Metaverse Services

NOTE: The potential security requirements will be updated based on the study progress in SA6.

# 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 TR 23.700-21: "Study on Application enablement architecture for mobile metaverse services".

[3] 3GPP TS 22.156: "Mobile Metaverse Services; Stage 1".

[4] 3GPP TS 33.434: "Security aspects of Service Enabler Architecture Layer (SEAL) for verticals".

[5] 3GPP TS 33.122: "Security aspects of Common API Framework (CAPIF) for 3GPP northbound APIs"

[6] 3GPP TS 33.501: "Security architecture and procedures for 5G System"

[7] 3GPP TS 23.434: "Service Enabler Architecture Layer for Verticals (SEAL); Functional architecture and information flows"

[8] 3GPP TS 23.438: “Service Enabler Architecture Layer for Verticals (SEAL); Digital Assets”

[9] 3GPP TS 23.437: “Service Enabler Architecture Layer for Verticals (SEAL); Spatial map and Spatial anchors”

# 3 Definitions of terms, symbols and abbreviations

## 3.1 Terms

For the purposes of the present document, the terms given in 3GPP TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in 3GPP TR 21.905 [1].

**Digital Asset Identifier:** In the context of this TR, digital asset identifier is used to uniquely identify a digital asset across different mobile metaverse services.

## 3.2 Symbols

For the purposes of the present document, the following symbols apply:

<symbol> <Explanation>

## 3.3 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in 3GPP TR 21.905 [1].

<ABBREVIATION> <Expansion>

clause includes the overview applicable for the study.

# 4 Security assumptions

The following security assumptions are applied to the study:

- The application enabler architecture for mobile metaverse services as described in TR 23.700-21 [2] is taken into account.

Editor’s Note: alignment with TR 23.700-21 is FFS.

- The security architecture, requirements and procedures for SEAL as defined in TS 33.434 [4] are used as a baseline.

Editor's Note: Whether SA6 architecture options are based on SEAL is FFS.

- Digital Asset Identifier is used in this study to identify a digital asset associated with a user.

# 5 Key issues

Editor’s Note: This clause contains all the key issues identified during the study.

## 5.1 Key Issue #1: Authorization supporting spatial localization service

### 5.1.1 Key issue details

In clause 4.1 of TR 23.700-21 [2], enabler support for managing spatial anchors is documented as a key issue, with the open issue regarding the access to spatial anchor as the following:

"*How to discover spatial anchors by the consumer (e.g. UE, VAL server)?*"

In clause 4.4 of TR 23.700-21 [2], exposing spatial map to third parties is documented as a key issue, with the open issue regarding the third party who needs to be authorized as the following:

"*How to expose a spatial map to authorized third parties?*"

Either for discovering spatial anchors or for exposing spatial maps, authorization of the consumer (e.g. UE, VAL server) needs to be considered. This key issue focuses on the authorization aspect supporting spatial localization service.

### 5.1.2 Security threats

Spatial map or spatial anchor could be a piece of information sensitive to the operator or the operator’s customer or the users in the map. If the consumer (e.g. UE, VAL server) is not authorized for obtaining the spatial map or accessing the spatial anchor, such sensitive information could be leaked to an undesired party. Further, the operator will not be able to correctly charge the consumer (e.g. UE, VAL server) for using spatial localization service supporting localized mobile metaverse services.

Editor’s Note: What sensitive information is for an operator or operator’s customer is FFS.

### 5.1.3 Potential security requirements

The 5G system shall provide a means to authorize a consumer (e.g. UE, VAL server) for accessing spatial localization services (e.g. spatial map obtaining, spatial anchor accessing).

Editor’s Note: The requirement details with respect to the potential consumers of localized mobile metaverse services, the host of such service and exposed information via such service is subject to SA6 progress.

Editor’s Note: Whether the spatial map contains spatial anchors from other users and the potential resulting threats and requirements are FFS.

## 5.2 Key Issue #2: Privacy of user sensitive information

### 5.2.1 Key issue details

According to clause 4.2 of TR 23.700-21 [2], user sensitive information (e.g. relating to user/UE identity, body movement or location, authentication result) needs to be accessed, managed and exposed through the enabler layer for localized mobile metaverse service.

In clause 4.2 of TR 23.700-21 [2], exposure of user sensitive information is documented as a key issue.,

This key issue focuses on the privacy aspect of user sensitive information which is transferred within or outside the network.

### 5.2.2 Security threats

User sensitive information needs to be accessed and exposed through the enabler layer to a party other than the user. Without proper protection, the privacy sensitive information could be leaked to undesired party, leading to privacy violation, trust and reputation impairment, regulatory incompliance, etc. An attacker can avail the user sensitive information to launch targeted attacks that cause data breaches, identity theft, etc.

### 5.2.3 Potential security requirements

The 5G system shall provide a means for privacy protection of user sensitive information during exposure of user specific information (e.g. user identity, user location) in localized mobile metaverse services through the application enabler layer.

## 5.3 Key issue #3: Security aspects of digital asset container in 5G

### 5.3.1 Key issue details

Avatar and digital asset support, including digital asset avatar management and discovery, is discussed in Key issue #3 in TR 23.700-21[2]. Correspondingly, the security aspect of the digital asset as well as the digital asset container itself deserve a thorough KI in order to further clarify the potential risks of the digital asset container in 5G.

This KI aims to comprehensively study the security requirements of the digital asset container in 5G.

### 5.3.2 Security threats

An attacker may access a digital asset if the authentication and authorisation of the usage of digital assets are not performed.

### 5.3.3 Potential security requirements

The 5G system shall support to authenticate and authorize a digital asset service consumer to access the digital asset(s) in a digital asset container.

NOTE: Digital asset service can be consumed by VAL applications as per TR 23.700-21[2], e.g. VAL client, VAL server.

## 5.4 Key Issue #4: Authentication and authorization of digital representation

### 5.4.1 Key issue details

In clause 7.2.4 of TS 22.156 [3], the following requirement implies the need of authentication of digital assets:

"*[R-7.2.4-002] The 5G system shall provide mechanisms to certify the authenticity of digital assets associated with a user.*"

In clause 7.2.3 of TS 22.156 [3], the following requirement implies the need of authorization of digital assets:

"*[R-7.2.3-001] Subject to operator policy, regulatory requirements and user consent, the 5G system shall be able to authorize the avatar to be used in mobile metaverse services.*"

Digital assets used in mobile metaverse services can be digital representation (avatar), software licenses, gift certificates, tokens, etc., which should be uniquely identifiable according to the definition of in clause 3.1 of TS 22.156 [3]. Avatars are digital representations of users interacting with the metaverse and other users in mobile metaverse services. In current mobile network services, users need to be authenticated to connect to mobile networks and authorized to access the requested services. In mobile metaverse services with avatar representing the user, user authentication and authorization need to be realized via the avatar.

Avatar and digital asset support key issue (KI#3) and requirements were described in clauses 4.3 and 5.4 of TR 23.700-21 [2], and corresponding solutions (Solution #5, 6, 7, 9) were introduced in clauses 7.5, 7.6, 7.7 and 7.9 of the same TR. The KI was concluded for normative work based on Solution #5, Solution #6, Solution #7 and Solution #9. According to the KI and requirements:

Avatars are digital representations of users interacting with the metaverse and with other users. The application enabler layer can enable creation, discovery, and management of avatar profiles for users to offload applications and enable Core Network functionality across services and verticals. The metaverse enablement services provide mechanisms to create, update, get/discover avatars as digital assets.

According to the solutions for avatar support in TR 23.700-21 [2], metaverse is considered as a digital world which is a replica of a real world. Most of the metaverse applications need avatars for the users to interact with the applications. And for each application, users may require to create, modify, get and delete avatars. Each avatar (regardless of application) will have some common properties. Depending on the metaverse application from which the user is interested to take the service, he/she can choose his/her avatar and the related information when needed. Also, a user can move between metaverse applications using the same avatar seamlessly and taking into account the constraints of the visited application. The solutions in TR 23.700-21 [2] provide support to manage digital avatars for the users, e.g. create, update, get, delete, discover, upload, download avatar and link/subscribe avatar to user/subscriber, etc., for specific metaverse applications, based on different architecture assumptions. It's mentioned in some solutions that the security mechanisms for the solutions need to be studied, or the IEs to be determined in the normative phase need to take security into considerations.

This key issue focuses on authentication and authorization of digital representation (e.g. avatar) which has its unique identifier. E.g. the user or application client on behalf of the user should be authorized to get/download an avatar which represents the user for a specific application; the metaverse application server is capable to verify if the user or application client on behalf of the user is authorized to use the avatar to interact with the metaverse application server and validate the authenticity of the avatar.

### 5.4.2 Security threats

Without authentication of avatar, an attacker can falsify an avatar to impersonate the user represented by the legitimate avatar. E.g. an attacker may download avatars of other users or generate his/her own avatar by copy-paste other user's avatar, and use the avatar to represent him/herself when interacting with the metaverse and with other users. As long as the association between the avatar and the user/subscriber being represented by the avatar cannot be verified in a mobile metaverse service, such attack cannot be detected. Then the attacker can manipulate the falsified/copied avatar in a mobile metaverse service to launch more types of attacks. Even if the unique identifier of a legitimate avatar can be changed from time to time, the attacker can still launch such attack during the valid period of the identifier.

### 5.4.3 Potential security requirements

The 5G system shall provide a means to support authenticating a digital representation to represent a user in mobile metaverse services.

The 5G system shall provide a means to support authorizing the user/subscriber to use the digital representation (avatar) in mobile metaverse services.

NOTE:  User authentication is not in the scope of the study.

NOTE: User identification is out of scope of the 5GC.

## 5.X Key Issue #X: <Key Issue Name>

### 5.X.1 Key issue details

### 5.X.2 Security threats

### 5.X.3 Potential security requirements

# 6 Solutions

Editor’s Note: This clause contains the proposed solutions addressing the identified key issues.

6.0 Mapping of solutions to key issues

Editor's Note: This clause contains a table mapping between key issues and solutions.

**Table 6.0-1: Mapping of solutions to key issues**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Solutions** | **KI#1** | **KI#2** | **KI#3** | **KI#4** |
| **Solution #1** | X |  |  |  |
| **Solution #2** | X |  |  |  |
| **Solution #3** | X |  |  |  |
| **Solution #4** |  | X |  |  |
| **Solution #5** |  | X |  |  |
| **Solution #6** |  |  | X |  |
| **Solution #7** |  |  |  | X |
| **Solution #8** |  |  | X |  |
| **Solution #9** |  |  | X |  |
| **Solution #10** |  |  |  | X |
| **Solution #11** |  |  |  | X |

## 6.1 Solution #1: Support for spatial localization service authorization

### 6.1.1 Introduction

This solution is for KI #1 and addresses the security requirements for authorizing UE to access spatial localization services. This solution is based on the SEAL identity management (SIM) service to perform UE authentication and authorization.

### 6.1.2 Solution details

Before getting authorization to specific service, the VAL UE authentication is executed by the SIM-S as described in TS 33.434 [4]. After successful authentication, the SIM-C requests and receives an access token from SIM-S as shown in Figure 6.1.2-1.



**Figure 6.1.2-1: Get Access Token**

1. User Authentication is completed between VAL UE and the SIM-S.
2. The VAL UE sends an access token request to the SIM-S, including the identity of the VAL UE and the specific spatial localization service the UE requests to access.
3. The SIM-S authorizes the VAL UE for the requested service and provides access token for the VAL UE.

With the received access token, the VAL UE can request for spatial localization service from corresponding SEAL server supporting the requested service. The procedure of getting spatial map for metaverse application is shown in Figure 6.1.2-2.



**Figure 6.1.2-2: Get Spatial Map**

1. A secure channel is established between SEAL client and SEAL server. Subsequent communication makes use of this channel.

1. The VAL UE sends a request message containing the access token to the SEAL server to get the spatial map via SEAL LM client.

2. On receiving the service authorization message, the SEAL server validates the access token.

3. If the access token is valid, the SEAL server provides the spatial map information to the VAL UE via SEAL client. Otherwise, the response included the failure cause indicating that the token is invalid.

The same procedure can also be applied for getting spatial anchor and any other spatial localization services provided by other SEAL servers by changing the request service.

NOTE: The SEAL server(s) supporting spatial anchor management services, spatial map management services, and other possible spatial localization services are aligned with TR 23.700-21[2].

### 6.1.3 Evaluation

This solution satisfies the potential security requirements of authorizing UE for access spatial localization services(e.g. spatial map obtaining, spatial anchor accessing).

In particular, the SIM-S performs UE authorization by issuing UE access token for the requested spatial localization service. The access token is included in the service request sent by UE and verified by the corresponding SEAL server. If the verification is successful, the UE will be granted use of the requested spatial localization service.

To achieve this, the following changes are needed:

- Spatial localization related SEAL service IDs are assigned and provisioned to the SIM-S.

- A new SEAL server for spatial anchor management is needed to provide functionalities for UE authorization checking.

- A new SEAL server for spatial map management is needed to provide functionalities for UE authorization checking.

## 6.2 Solution #2: Solution for KI#1 on Authorization supporting spatial localization service with CAPIF Core Function (CCF)

### 6.2.1 Introduction

The solution addresses KI#1 on Authorization supporting spatial localization service.

According to TR 23.700-21[2], SEAL architecture as defined in TS 23.434[7] is reused for spatial anchor and spatial map management. Based on TS 33.434[4], when CAPIF is used as specified in TS 23.434[7], the security mechanism for CAPIF specified in TS 33.122[5] shall be followed.

The solution proposes security procedures to authenticate and authorize the spatial anchor management service consumers.

### 6.2.2 Solution details

In TR 23.700-21[2], Service Enabler Architecture Layer (SEAL) service is enhanced to provide spatial anchor management service to the VAL server (VAL-S) or SEAL client (SEAL-C) in UE. VAL-C/SEAL-C may invoke service provided by SEAL server (SEAL-S) to create, read, update, delete or discovery spatial anchor. Based on TS 23.434[7], when CAPIF is used, the VAL server acts as CAPIF's API invoker and the SEAL server acts as CAPIF's API exposing function. CAPIF framework is reused in this solution to authorize spatial localization service consumer.

NOTE: OAuth 2.0 token based authorization of CAPIF is adopted for this solution.

#### 6.2.2.1 Procedure of authorization for spatial localization service



**Figure 6.2.2.1-1: Procedure of authorization for spatial localization service**

0. Mutual authentication is performed between VAL server and CAPIF Core function (CCF), and secure session is established between the entities.

1. VAL server sends request to CCF to receive access token for authorization to create/read/update/delete (CRUD) or discovery spatial anchor(s) in SEAL server. The request specifies the desired operation, e.g. CRUD, spatial anchor instances to be operated, etc.

2. The CCF verifies the request's based on preconfigured policies and issues an access token to the VAL server if authorized. This token incorporates authorization attributes in different access levels.

For example,

- SEAL/VAL service level: e.g. permission to CRUD operations on metaverse localization services such as spatial anchor management service, spatial map management service.

- Spatial anchor instance level: permission to operate on specific spatial anchor instance(s)

- 3rd party service level: e.g. if multiple service information (from different service provider) is included in a spatial anchor, authorization to update all or specific service information associated with a spatial anchor.

3. The CCF returns the access token to the VAL server.

4. After mutual authentication, the VAL server establishes secure session with a SEAL server.

5. The VAL Server, equipped with the access token, sends CRUD spatial anchor request to the SEAL Server.

6. The SEAL Server verifies the access token, e.g. the issuer (CCF), subject (VAL Server), and scope (matching the requested operation).

7-8. If all validations and checks pass, the SEAL Server processes the CRUD operation on spatial anchors and sends a success response to the VAL Server.

NOTE: The security procedure is applicable to spatial map management with changing spatial anchor to spatial map.

### 6.2.3 Evaluation

The solution addresses requirements of Key issue #1 to support authorization of a consumer (e.g. UE, VAL server) for accessing spatial localization services (e.g. spatial map obtaining, spatial anchor accessing) based on CAPIF security mechanism.

According to TS 23.434 [7] and TS 23.437 [9], Spatial Anchor/Map Client and Val Server play the role of CAPIF API Invoker, Spatial Anchor/Map plays role of CAPIF AEF, hence CAPIF API Invoker, AEF and Core Function need to enhance to support Spatial Anchor/Map specific authorization.

The solution is aligned with Localized mobile metaverse services defined in TS 23.437 [9] and CAPIF.

## 6.3 Solution #3: Solution for KI#1 on Authorization supporting spatial localization service with CAPIF Core Function (CCF) in case of different resource owner

### 6.3.1 Introduction

The solution addresses KI#1 on Authorization supporting spatial localization service.

In Solution #2 of TR 23-700-21, Support for spatial anchor management of 23-700-2, a VAL server may include following information in the request when create a spatial anchor:

- service information of the product to associate it with the spatial anchor,

- access control rules defining which entities are permitted to discover and access the spatial anchor,

- customer premise information (e.g. a residence, office, or shop).

- spatial anchor discoverable visibility levels like universal to facilitate shared spatial anchor discovery

When one VAL server discovers or manages a spatial anchor created by another VAL server, the access control policies from the creating VAL server should be considered to authorize the spatial anchor services to the accessing VAL server.

The solution proposes to authorize one VAL server/SEAL client to access spatial anchor created by another VAL server/SEAL client with considering the access control polices from the creating VAL server.

CAPIF framework is based to authorize spatial localization service consumer in this solution.

NOTE: OAuth 2.0 token based authorization of CAPIF is adopted in this solution.

### 6.3.2 Solution details

A VAL server (VAL server1) creates a spatial anchor in a SEAL server which including access control rules/authorization policies in the creation request and the SEAL server synchronizes authorization policies with CAPIF Core Function (CCF) together with the VAL server1 information. When another VAL server (VAL server 2) requests access token from CCF for the spatial anchor, CCF checks the authorization policy of the spatial anchor. If VAL server2 is allowed to access the spatial anchor, CCF generates an access token and provides it to VAL server2. The VAL server2 initiates a spatial anchor service request along with the access token to the SEAL server. The SEAL server, upon successful validation of access token, provides the spatial anchor service to the VAL server2.

#### 6.3.2.1 Procedure of authorization of spatial anchor service with multiple VAL server



**Figure 6.3.2.1-1: Procedure of authorization of spatial anchor service with multiple VAL server**

0. VAL server1 and server2 are onboarded to CCF and authenticated with CCF. VAL server1 is authorized by CCF and access token for creation of spatial anchor has been obtained from CCF.

1. VAL server1 sends a spatial anchor creation request to SEAL server with the spatial anchor information and access token obtained from CCF.

2. SEAL server validates the access token against the service request and creates a spatial anchor, e.g. with spatial anchor id set to “anchor\_123”.

3. SEAL server sends spatial anchor creation response to VAL server1.

4. SEAL server publishes the spatial anchor information, e.g. spatial anchor id (anchor\_123), owner of the spatial anchor (VAL server 1) and optionally the authorization policies associated with the spatial anchor to CCF.

5. CCF stores the spatial anchor information in the VAL server1 profile.

6. VAL server 2 sends access token request to CCF to access the spatial anchor (anchor\_123).

7. CCF authorizes the request based on local policies (e.g. if a VAL server is allowed to consume spatial anchor related services) and authorization policies associated with the spatial anchor (e.g. if the anchor\_123 can be read/updated by the VAL server 2) presented in VAL server 1 profile. If VAL server2 is authorized, generates an access token.

NOTE: Optional CCF may implicitly ask authorization from VAL server1 if there's no authorization information associated with the spatial anchor based on local policies.

8. CCF sends the access token to VAL server2.

9. VAL server2 sends request to SEAL server to access the spatial anchor, along with the access token received from CCF.

10. SEAL server validates the access token and service request and perform the request on successful validation.

11 SEAL server sends the spatial anchor response to VAL server2.

Editor’s Note: Alignment with CAPIF is FFS.

### 6.3.3 Evaluation

The solution addresses requirements of Key issue #1 to support authorization of a consumer (e.g. UE, VAL server) for accessing spatial localization services (e.g. spatial map obtaining, spatial anchor accessing) based on CAPIF security mechanism, especially when the consumer of the spatial map/anchor is different to the owner of the spatial map/anchor.

According to TS 23.434 [7] and TS 23.437 [9], Spatial Anchor/Map Client and Val Server play the role of CAPIF API Invoker, Spatial Anchor/Map plays role of CAPIF AEF, hence CAPIF API Invoker, AEF and Core Function need to enhance to support Spatial Anchor/Map specific authorization, especially in the case that the consumer of the spatial map/anchor is different to the owner of the spatial map/anchor.

The solution is aligned with Localized mobile metaverse services defined in TS 23.437 [9].

Editor’s Note: Alignment with CAPIF is FFS.

## 6.4 Solution #4: Privacy protection for user sensitive information exposure

### 6.4.1 Introduction

This solution addresses Key Issue #2 on privacy of user sensitive information. Specifically, it addresses the requirements that the 5G system shall provide a means for privacy protection of user sensitive information during exposure of user specific information in localized mobile metaverse services through the application enabler layer.

### 6.4.2 Solution details

According to TR 23.700-21 [2], user sensitive information needs to be exposed through the enabler layer to a party other than the user. The information that can be exposed may include user/UE identity, body movement or location, ownership rights of digital assets, authentication result, etc. As all these information is related to a specific user, user privacy needs to be ensured when exposing any of the information, for which user authorization or user consent is required.

To protect the privacy of the user whose sensitive information is to be exposed, regardless of whether the user is a 3GPP subscriber or not, it is proposed to reuse RNAA framework defined in TS 33.122 [5], by regarding the user sensitive information as the resource of the user. The API invoker requesting the user sensitive information through the enable layer can only be authorized if the authorization function (CCF) obtains permission from the user, i.e. the resource owner.

For the case where the user is a 3GPP subscriber, the user consent framework defined in TS 33.501 [6] Annex V can also be reused for privacy protection via static user authorization based on UE subscription.

### 6.4.3 Evaluation

This solution reuses the existing security mechanisms to fulfill the requirements in KI#2 on privacy of user sensitive information. The existing mechanism can be based on either the procedure for RNAA (Resource owner-aware Northbound API Access) defined in TS 33.122 [5] or the user consent framework defined in TS 33.501 [6] Annex V if the user is a 3GPP subscriber. Hence no new security mechanism needs to be defined for KI#2.

## 6.5 Solution #5: Privacy protection during metaverse service discovery

### 6.5.1 Introduction

This solution addresses Key Issue #2 on privacy of user sensitive information.

According to 4.2 Key Issue #2: Exposure of user sensitive information of TR 23.700-21, ensuring appropriate user consent has been obtained is a critical aspect when handling sensitive information relating to or collected from a user, their devices or the applications installed at their devices. For instance, with the expected capability to access, manage and expose user specific avatar related information through the enabler layer it is of utmost importance to capture the consent of the user.

Spatial anchor, spatial map discovery are supported in solution#1 (clause 7.1) and solution#8 (clause 7.8) of TR 23.700-21 [2], this contribution proposes a solution to check user consent in discovery procedure, and return list of spatial anchors, spatial maps, which match user consent, to the metaverse service consumer or SEAL client.

### 6.5.2 Solution details



**Figure 6.5.2-1: Procedure of privacy protection during metaverse service discovery**

Precondition:

A list of spatial anchors, spatial maps and avatars are created, personal data required to support each spatial anchor, spatial map or avatar is registered.

The SEAL server is deployed in operator domain and trusted by the 3GPP core network.

1. UE initiates a discovery request to SEAL server to get the list of spatial anchors or spatial maps, AF specific UE Identifier, e.g. GPSI, is included in the request.

2. SEAL server requests the UDM directly to get the user consent for metaverse services based on the GPSI.

3. UDM sends the user consent for the metaverse services to the SEAL server.

Editor’s Note: whether the user consent information in the UDM can be specific for metaverse services is FFS.

4. SEAL server processes the user consent along with the personal data required to support registered spatial anchors or spatial maps.

5. SEAL server sends a list of spatial anchors or spatial maps which personal data requirements satisfy UE consent.

Editor’s Note: Whether user consent or resource owner authorization is most suitable to be used in this solution is FFS.

### 6.5.3 Evaluation

The solution addresses requirements of Key issue #2 to support privacy protection of user sensitive information during exposure of user specific information (e.g. user identity, user location) in localized mobile metaverse services through the application enabler layer.

UDM needs to enhance to support user consent retrieving and checking for protection of user sensitive information in metaverse services.

The solution assumes that the SEAL server is deployed in operator domain and trusted by the 3GPP core network, hence it can get user consent as specified in Annex V of TS 33.501.

Editor’s Note: Whether user consent or resource owner authorization is most suitable to be used in this solution is FFS.

Editor’s Note: whether the user consent information in the UDM can be specific for metaverse services is FFS.

## 6.6 Solution #6: Digital asset request validation

### 6.6.1 Introduction

This solution addresses key issue#3. In this solution it is assumed that the SEAL security procedure is re-used for user authentication and authorization as specified in 5.2 of TS 33.434 [4].

In this solution, it is proposed that the SEAL Server (Digital Asset Container Management) digitally signs the requested avatar object using the private key, obtained as part of key provisioning procedure defined in TS 33.434 [4].

#### 6.6.2.1 Access token request/response



**Figure 6.6.2.1-1: Updated VAL user authentication from TS 33.434 [4]**

Figure 6.6.2.1-1 describes the VAL Authentication Framework using the OpenID Connect protocol when using HTTPS as specified in TS 33.343 [4]. Additionally, at step 5a the SIM-S gets the Avatar/digital asset consumer list from the SEAL server (DACM) to verify the mapping between allowed user and the metaverse services based on the digital asset profile.

NOTE: Creation of user list or the asset or avatar profile in SEAL Server (DACM) is not in scope of this solution.

#### 6.6.2.2 Solution details



**Figure 6.6.2.2-1: Digital asset request validation**

1. The VAL Client/SEAL Client/VAL Server sends an avatar or digital asset download request to the SEAL Server (DACM) function with the Avatar ID, GPSI/External ID of the UE. The request also includes the access token to authorize the requestor.

2. The SEAL Server (DACM) function checks the authorization of the VAL Client/SEAL Client/VAL Server based on the Avatar ID, GPSI/External ID of the UE present in the request message against the allowed user list locally stored to perform the operation. If successful, the SEAL Server (DACM) function performs the media adaptation as per the request on the avatar object/media.

3. The SEAL server function sends a response to the VAL Client/SEAL Client/VAL Server indicating success or failure of the operation. If successful, the avatar object/media/base avatar is included in the response, the SEAL Server (DACM) returns digitally signed avatar object/media/base avatar using the keys obtained from SEAL KM Server. The VAL Client/SEAL Client/ VAL Server which sent the download request if in possession of the required public key would be able to verify the digital signature in the avatar object/media/base avatar for its use.

6.6.3 Evaluation

This solution addresses the security requirements of key issue#3. In this solution it is assumed that the SEAL security procedure is re-used for authentication and authorization as specified in 5.2 of TS 33.434 [4].

In this solution, it is proposed that the SEAL Server (Digital Asset Container Management) digitally signs the requested avatar object using the private key, obtained as part of key provisioning procedure defined in TS 33.434 [4].

Editor’s Note: Further evaluation is FFS.

## 6.7 Solution #7: Authorize avatar by metaverse service provider

### 6.7.1 Introduction

This solution addresses Key Issue #4 on Authentication and authorization of digital representation.

Avatar and digital asset support, including digital asset avatar management and discovery, is discussed in Key issue #3 in TR 23.700-21[2]. The corresponding requirements were defined in clause 5.4 of the TR 23.700-21[2] which include, for example, subject to operator policy, regulatory requirements and user consent, the metaverse enablement service shall provide digital asset management mechanisms as follows:

- to create, update, retrieve, delete and discover digital assets securely.

- to manage associations between digital assets and user identifiers.

- to allow an authorized third party to manage digital asset(s) associated with a user.

Correspondingly, KI#4 on Authentication and authorization of digital representation was described in TR 33.721 which requires 5G system support authenticating a digital representation to represent a user in mobile metaverse services and authorizing the VAL user to use the digital representation (avatar) in mobile metaverse service.

Architecture, procedures and information flows for digital asset service to support mobile metaverse services are specified in TS 23.438 [8]. Users can be associated with one or more digital assets like Avatars, software licenses, files, etc. Applications like mobile metaverse services can utilize the digital assets related to users, and the users can benefit from having the use of their digital assets between the various metaverse applications/platforms in an interoperable way.

In the current release of this specification, the digital asset type is limited to digital avatar.

This solution aims to support metaverse service provider to authorize the VAL user to use the avatar when the users have the use of their avatars between the various metaverse service providers.

### 6.7.2 Solution details



**Figure 6.7.2-1: Procedure to authorize usage of avatar in metaverse service**

Precondition:

- Avatar is created and stored in digital asset server which included authorization information provisioned by the owner of the digital asset as part of DA profile, which includes owner list, allowed user list, allowed application list, spatial conditions, expiry time, etc., refer to clause 7.2 of TS 23.438 [8] for the detail authorization information in DA profile.

- Digital asset (DA) client of the VAL UE retrieves avatar from the DA server.

- VAL client of the VAL UE retrieves avatar from the DA client inside UE via DA-C interface which is out of scope of 3GPP.

NOTE : Assume only one VAL user is supported for a VAL UE, which is corresponding to the subscriber of the UE and the VAL user id can be mapped to the VAL UE Id or GPSI.

Procedure:

1. The VAL client, on behalf of the VAL user, sends metaverse service request to a metaverse service provider including VAL user id, avatar and avatar id.

2. The metaverse service provider sends request to DA server to check if the VAL client on behalf of the VAL user is allowed to use the avatar, the request includes avatar/DA id, VAL user id, application id of the metaverse service provider, location of the VAL client or metaverse service provider, usage time.

3. The DA server retrieves DA profile based on DA id, and checks if the VAL user id is in the allowed user id list, the application id(s) is in the allowed application list, the location is matched spatial conditions, and the usage time is before expiry time of the DA/avatar.

4. The DA server returns the checking result to the metaverse service provider.

5. Metaverse service provider proceeds with the metaverse service request, e.g. show the avatar in spatial anchor/map, and sends response to the DA client.

### 6.7.3 Evaluation

The solution addresses requirements of Key issue #4 to support authorizing the VAL user to use the digital representation (avatar) in mobile metaverse service.

SEAL server, e.g. DA server, needs to enhance to support authorization request from other VAL/SEAL server such as spatial anchor/map server.

The solution assumes only one VAL user is supported for a VAL UE, which is corresponding to the subscriber of the UE and the VAL user id can be mapped to the VAL UE Id or GPSI.

## 6.8 Solution #8: Authenticate and authorize DA client to create a digital asset

### 6.8.1 Introduction

This solution addresses Key Issue #3 on Security aspects of digital asset container in 5G.

Avatar and digital asset support, including digital asset avatar management and discovery, is discussed in Key issue #3 in TR 23.700-21[2]. The corresponding requirements were defined in clause 5.4 of the TR 23.700-21[2] which include, for example, subject to operator policy, regulatory requirements and user consent, the metaverse enablement service shall provide digital asset management mechanisms as follows:

- to create, update, retrieve, delete and discover digital assets securely.

- to manage associations between digital assets and user identifiers.

- to allow an authorized third party to manage digital asset(s) associated with a user.

Correspondingly, KI#3 on Security aspects of digital asset container in 5G were described in TR 33.721 which requires 5G system to authenticate and authorize a digital asset service consumer to access the digital asset(s) in a digital asset container.

Architecture, procedures and information flows for digital asset service to support mobile metaverse services are specified in TS 23.438 [8]. Especially, figure 6.2-2 of TS 23.438 [8] illustrates the DA architecture to support Metaverse services using CAPIF architecture.

Users can be associated with one or more digital assets like Avatars, software licenses, files, etc. Applications like mobile metaverse services can utilize the digital assets related to users, and the users can benefit from having the use of their digital assets between the various metaverse applications/platforms in an interoperable way.

This solution aims to authenticate and authorize a digital asset client, on behalf of a VAL user, to create a digital asset based on CAPIF.

### 6.8.2 Solution details



**Figure 6.8.2-1: Procedure to authorize digital asset client for DA creation**

Precondition:

- Digital asset client (DA client/API Invoker), CAPIF core function (CCF), digital asset DA server (DA server/AEF) are preconfigured with certificates and trust anchor for TLS based mutual authentication.

- CAPIF core function (CCF) is preconfigured with authorization policies for API Invokers.

- The user behind the UE has been identified and authenticated by the DA server.

Editor’s Note: How DA server authenticates the user is FFS.

Procedure:

1. DA client, on behalf of subscriber user, is onboarded successfully and CAPIF-1E authentication is performed with CCF.

2. DA client sends Access Token Request to CCF for create digital asset service, the request includes client id, optional digital asset type (e.g. avatar).

3. CCF checks if the DA client is allowed to create related digital asset based on client/API invoker profile and corresponding authorization policies.

4. If the DA client is allowed to create digital asset, CCF sends access token to the DA client which includes authorized operations for the DA client.

5. After received access token for creation of digital asset, the DA client performs CAPIF-2E authentication with DA server.

6. DA client sends DA creation request to DA server which includes access token besides other parameters such as client id, user id, digital asset type (e.g. avatar) and digital asset profile parameters.

7. DA server verifies the access token.

8. If successfully verified the access token, DA server creates digital assets according to DA profile, which includes DA type (e.g. avatar), authorization information (e.g. owner user which is set to the user id in the request, allowed applications, allowed users, Metaverse service providers, location, expire time, etc.).

9. DA server returns digital asset identifier to the DA client.

### 6.8.3 Evaluation

The solution addresses requirements of Key issue #3 to support authorizing DA client to manage digital asset container in 5G.

DA server, DA client, CCF need to enhance to support authorization of DA creation.

## 6.9 Solution #9: Authenticate and authorize VAL server/client to access digital asset

### 6.9.1 Introduction

This solution addresses Key Issue #3 on Security aspects of digital asset container in 5G.

Avatar and digital asset support, including digital asset avatar management and discovery, is discussed in Key issue #3 in TR 23.700-21[2]. The corresponding requirements were defined in clause 5.4 of the TR 23.700-21[2] which include, for example, subject to operator policy, regulatory requirements and user consent, the metaverse enablement service shall provide digital asset management mechanisms as follows:

- to create, update, retrieve, delete and discover digital assets securely.

- to manage associations between digital assets and user identifiers.

- to allow an authorized third party to manage digital asset(s) associated with a user.

Correspondingly, KI#3 on Security aspects of digital asset container in 5G was described in TR 33.721 which requires 5G system to authenticate and authorize a digital asset service consumer to access the digital asset(s) in a digital asset container.

Architecture, procedures and information flows for digital asset service to support mobile metaverse services are specified in TS 23.438 [8]. Especially, figure 6.2-2 of TS 23.438 [8] illustrates the DA architecture to support Metaverse services using CAPIF architecture.

Users can be associated with one or more digital assets like Avatars, software licenses, files, etc. Applications like mobile metaverse services can utilize the digital assets related to users, and the users can benefit from having the use of their digital assets between the various metaverse applications/platforms in an interoperable way.

This solution aims to authenticate and authorize a VAL client (a.k.a DA client) or VAL server to access digital asset(s) based on CAPIF.

### 6.9.2 Solution details



**Figure 6.9.2-1: Procedure to authorize VAL\_C/VAL\_S to access digital asset**

Precondition:

- Digital asset client (DA client/API Invoker), CAPIF core function (CCF), digital asset DA server (DA server/AEF) are preconfigured with certificates and trust anchor for TLS based mutual authentication.

- CAPIF core function (CCF) is preconfigured with authorization policies for API Invokers, which may define authorization policies that what AEF services can be accessed by the API Invokers.

- Digital asset is created and stored in DA server which may include authorization information provisioned by the owner of the digital asset, refer to clause 7.2 of TS 23.438 [8] for the detail information in DA profile.

Procedure:

1. VAL server/client (or API invoker) is onboarded successfully and CAPIF-1E authentication is performed with CCF. The authorization polices for the API invoker may be updated based on the API invoker profile.

2. VAL server/client sends Access Token Request to CCF to update/retrieve/delete digital asset, the request includes client id, and optional digital asset identifier.

3. CCF checks if the VAL server/client is allowed to access digital asset services based on preconfigured authorization policies for the API invoker or updated policies in step 1.

4. If so, based on the deployment option, CCF may check if authorization information associated to the digital asset identifier existed locally. If not, CCF locates corresponding DA server based on API type, location, UE information, digital asset identifier, etc., and performs CAPIF-3 authentication with the DA server.

5. Optionally, CCF sends request to the DA server to get authorization information of the digital asset.

6. DA server returned authorization information of the digital asset based on the DA profile created in precondition step.

7. CCF stores the authorization information of the digital asset, and checks if the VAL server/client, is allowed to update/retrieve/delete the digital asset based on allowed application list.

8. Based on the policy, CCF may ask authorization from the resource owner based on RNAA framework in case the resource owner is associated with a UE which can be identified by CCF.

Editor’s Note: How CCF identifies the UE based on resource owner user id is FFS.

9. If the VAL server/client is allowed to update/retrieve/delete digital asset, CCF sends access token to the VAL server/client which includes authorized operations and parameters.

10. After received access token for accessing of digital asset, the VAL server/client performs CAPIF-2E authentication with DA server.

11. VAL server/client sends DA update/retrieve/delete request to DA server which includes access token besides other parameters.

12-13. DA server verifies the access token and proceeds the request accordingly and returns the result to the VAL server/client.

Note: Instead of retrieving authorization information of the DA from DA server in step 4 and 8, the CCF may only authorize the VAL server/client based on preconfigured/updated local authorization policies and include related permissions in the scope of the access token, and DA server may further authorized the access based on authorization information of the required DA.

### 6.9.3 Evaluation

The solution addresses requirements of Key issue #3 to support authorizing DA client to manage digital asset container in 5G.

DA server, DA client, CCF need to enhance to support authorization of DA access. Especially DA server enhances to enable CCF to retrieve authorisation policies of a digital asset.

RNAA is reused for getting authorization from the resource owner in case the resource owner is associated with a UE which can be identified by CCF

Editor’s Note: How CCF identifies the UE based on resource owner user id is FFS.

## 6.10 Solution #10: Avatar authentication and authorization supported by A-DACM architecture

### 6.10.1 Introduction

This solution addresses KI #4 on authentication of digital representation.

In this solution, avatar authentication and authorization are achieved via the tokens (ID token, access token) using OpenID Connect 1.0 and OAuth 2.0, in the similar way as VAL user authentication defined in TS 33.434 [4]. The ID token is consumed by the VAL client in the UE for authenticating the avatar used by the user. The access token is consumed by the VAL server for authenticating and authorizing the avatar sent by the VAL client for accessing the requested metaverse service.

### 6.10.2 Solution details

#### 6.10.2.1 Description

The solution assumes the following:

- Based on solution #6, a user has been authenticated as specified in clause 5.2 of TS 33.434 [4] for downloading his/her avatar from the A-DACM function to the UE he/she is using. The downloaded avatar in the UE can be shared by multiple metaverse VAL clients in the UE.

- Based on solution #6, avatar object is digitally signed by the A-DACM function, which creates and manages the avatar, using its private key.

- The SIM-S is configured with the certificate of the A-DACM function. The SIM-S checks the authenticity of the avatar object if sent by the VAL client with the certificate before providing the tokens (ID token, access token) to the VAL client.

- SIM-S generates the tokens based on the checking results from the A-DACM function storing the avatar profile, which contains the information of the linkage between avatar and user(s), and the mapping between avatar and allowed metaverse services.

- The VAL UE and VAL server are configured with the certificate of the SIM-S.

#### 6.10.2.2 Avatar authentication procedure

 **Figure 6.10.2.2-1: Avatar Authentication Procedure for Metaverse VAL Client**

0. VAL UE establishes a secure tunnel with the SIM-S.

1. When the log-in user on the UE intends to access a metaverse service via the corresponding metaverse VAL client in the UE by using the downloaded avatar to represent him/her, the VAL client in the UE sends an OpenID Connect Authentication Request to the SIM-S. The request contains the VAL client ID, User ID, Avatar ID and optionally the corresponding Avatar Object, and the ID/name of the metaverse service to be requested. The avatar object is digitally signed by the A-DACM function.

2. The SIM-S verifies the authenticity of the received avatar object using the certification of the A-DACM function if avatar object is included in the request message.

3a. If the verification is successful, the SIM-S sends a Representation Check Request to the A-DACM function. The request contains at least the User ID, Avatar ID and service ID/name.

3b. If the verification fails, the SIM-S regards the received avatar as unauthentic and returns a response to the VAL client in the UE, indicating verification failure of avatar object.

4. Based on the received User ID and Avatar ID, the A-DACM function determines whether the avatar is allowed to be used by the user by checking against the avatar profile. The A-DACM function also determines whether the avatar is allowed to access the service indicated by the service ID/name.

5. The A-DACM function returns the Representation Check Response with the check results and optionally additional information e.g. the ID of the VAL Server providing the requested service, an indication on whether the avatar can be shared by multiple metaverse services.

6a. If the check results are positive, the SIM-S sends an OpenID Connect Authentication Response to the VAL client in the UE containing an authorization code and the additional information obtained from the A-DACM function if any.

6a. If any of the check results is negative, the SIM-S sends an OpenID Connect Authentication Response to the VAL client in the UE, indicating the mismatch between the avatar and user or between the avatar and service.

7. Upon receiving the authorization code, the VAL client in the UE sends an OpenID Connect Token Request to the SIM-S passing the code. The request also includes the VAL client ID or UE ID (e.g. GPSI) and VAL server ID.

8. The SIM-S sends an OpenID Connect Token Response to the VAL client containing an ID-Token\_A and an access token.

The ID-Token\_A contains at least the following claims:

- Issuer: the SIM-S

- Subject: Avatar ID || User ID (i.e. concatenation of user ID and avatar ID)

- Audience: the VAL client ID or the UE ID

NOTE: If the avatar can be shared by multiple metaverse services, the audience contains the UE ID. The ID-Token\_A is then stored in a common secure environment in the UE that can be shared to multiple VAL clients. The next activated VAL client will then check whether the locally stored ID-Token\_A can be used without sending OpenID Connect Authentication Request to the SIM-S as in step #1.

- Expiration Time: the expiration time after which the token must not be accepted for processing

- Optional additional claims if necessary, e.g. avatar object (e.g. image, media), etc.

9. The ID-Token\_A is consumed by the metaverse VAL client or VAL UE, which processes the following steps:

- verify the integrity of the token using the certificate of the SIM-S

- check whether the avatar ID and user ID as the value of subject claim match the identities of the requesting user and the avatar representing the user

- check whether the audience claim matches the identity of the VAL client itself or VAL UE

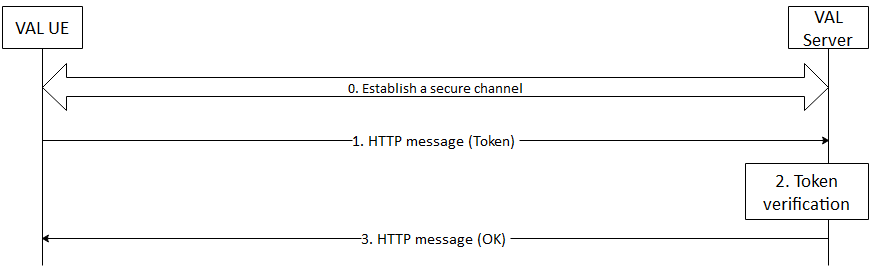
- check whether the token is not expired

- optionally check the additional claims if any, e.g. whether the avatar object in the claim matches the avatar object selected by the user

By checking the claims in the obtained ID-Token\_A, the metaverse VAL client is able to determine that the avatar object used by the user is authentic and allowed to be used by the requesting user.

The access token returned by the SIM-S is used by the VAL client for service access with the avatar object as described in clause 6.10.2.3.

#### 6.10.2.3 Avatar authorization procedure

****

**Figure 6.10.2.3-1: Avatar Authorization Procedure for Metaverse Service Access**

0. A secure HTTP tunnel between VAL client and VAL server is established.

1. The VAL client sends an HTTP message to the VAL server containing the Avatar ID and Object, the service ID/name and the access token. The access token contains at least the following claims:

- Issuer: the SIM-S

- Subject: Avatar ID || User ID (i.e. concatenation of avatar ID and user ID)

- Audience: the VAL Server ID of the metaverse service requested by the user

NOTE: the VAL Server ID can be sent to the SIM-S by the VAL client at step #7 in clause 6.10.2.2 or by the A-DACM function at step #5 in clause 6.10.2.2.

- Scope: service ID/name

- Expiration Time: the expiration time after which the token must not be accepted for processing

- Additional claims if necessary, e.g. avatar object (e.g. image, media), etc.

2. The VAL server authorizes the avatar for accessing the requested services only if the access token is valid.

The access token is consumed by the metaverse VAL server, which processes the following steps:

- verify the integrity of the token using the certificate of the SIM-S

- check whether the avatar ID and user ID as the value of subject claim match the identities of the requesting user and the avatar representing the user

- check whether the audience claim matches the identity of the VAL server itself

- check whether the scope claim matches the identity of the requested service

- check whether the token is not expired

- check the additional claims if any, e.g. whether the avatar object in the claim matches the avatar object sent by the VAL client

After successful verification of the access token, the VAL server is able to determine that the avatar object sent by the VAL client is authentic and allowed to be used by the requesting user (i.e. implicit avatar authentication) and is allowed to access the requested metaverse (avatar authorization).

3. The VAL server responds the request with OK or failure. The VAL server may also provide service related information to the VAL client.

### 6.10.3 Evaluation

This solution fulfills the requirements in KI#4 on authenticating a digital representation (avatar) to represent a user and authorizing the user/subscriber to use the digital representation in mobile metaverse services.

ID token based on OpenID Connect 1.0 is used for avatar authentication. The information that can be used for authenticating an avatar to represent a user is the binding of avatar ID and user ID in the subject claim of the ID token for avatar.

Access token based on OAuth 2.0 is used for user authorization. The information that can be used for authorizing a user to use the avatar is the binding of avatar ID and user ID in the subject claim of the access token.

The entity, which is able to verify the authenticity of the avatar object (e.g. SIM-S), is responsible for generating ID token and access token. The entity binds avatar ID and user ID in the tokens based on the linkage information between avatar and user as well as the mapping between avatar and allowed metaverse services, which is stored in the avatar profile maintained by the A-DACM function.

NOTE: It is assumed that user ID in this solution can refer to VAL user ID defined in TS 23.434 [7] clause 7.

The pre-requisite of both avatar authentication and user authorization is that the user is already authenticated as specified in clause 5.2 of TS 33.434 [4].

This solution mainly impacts the SIM-S, which needs to include the binding information of the avatar and user in the tokens, and A-DACM function which needs to interface with the SIM-S. Avatar object is optionally included in the ID token and access token if downloaded in the UE.

## 6.11 Solution #11: Avatar authentication and authorization supported by CAPIF

### 6.11.1 Introduction

This solution addresses KI #4 on authentication and authorization of digital representation.

When CAPIF is used to support metaverse enablement services, it serves as the framework for north bound API of the DA service provided by the DA server. In this case, the DA server can be mapped to the AEF and the API invoker in VAL UE can be mapped to the API invoker in CAPIF functional model.

For authenticating the avatar and authorizing its usage by the user, the API invoker in the VAL UE invokes the DA service API for retrieving DA profile from the AEF (i.e. DA server). Based on validation of avatar authenticity and the retrieved DA profile containing the association between avatar and user, the VAL UE is able to determine whether the avatar can represent the user in the application and whether the user is allowed to use the avatar in the application.

### 6.11.2 Solution details

1. The metaverse application triggers the VAL UE for avatar authentication and authorization by passing at least the avatar ID and corresponding avatar (media), VAL user ID, app ID.

As the avatar is created and signed by the DA server, the VAL UE first verifies the authenticity of the avatar by verifying the signature of the avatar using the certificate of the DA server if locally available. Only when the avatar is verified as authentic, the VAL UE will then proceed with the following steps.

****

**Figure 6.11.2-1: Avatar authentication and authorization**

2. The API invoker in the VAL UE sends a DA service API invocation request to the DA server, including at least invoker ID, identifier of DA service API to be invoked (i.e. SS\_DAProfileManagement\_Retrieve in TS 23.438 [8]), avatar ID and optionally app ID. If the VAL UE is not able to verify the authenticity of the avatar (e.g. no DA server certificate), it also sends the received avatar to the DA server for verification.

3. If avatar is included in the request, the DA server first verifies the authenticity of the avatar, and then checks the app ID if received. Only when the avatar is verified as authentic and allowed to be used in the application, the DA server will then accept the DA service API invocation request. Otherwise, the DA server returns a failure response to the VAL UE. If needed, the DA server may obtain more authorization information from the CCF before responding.

4. The DA server sends the DA service API invocation response to the VAL UE, which contains either the requested DA profile or a result.

5. If the requested DA profile or a result is received, it implies that the authenticity of the avatar is successfully verified, which means that avatar is implicitly authenticated. The VAL UE then determines whether the authentic avatar can be used to represent the VAL user or whether the VAL user is allowed to use the avatar, by checking the association between avatar and VAL user in the DA profile.

6. The metaverse application in the UE proceeds only if the avatar authentication and authorization is successful.

### 6.11.3 Evaluation

This solution addresses KI #4 on authentication and authorization of digital representation by applying the CAPIF framework.

From the perspective of the avatar, the combination of successful verification of avatar authenticity and identified association between the avatar and user can ensure that the avatar is authenticated to represent the user in a mobile metaverse application. From the perspective of the user, this combination also ensures that the user is authorized to use the avatar in a mobile metaverse application. Hence, both requirements in key issue #4 are fulfilled.

This solution reuses the procedures in CAPIF without impact on CAPIF or DA service architecture. The DA server or the VAL UE is required to verify the authenticity of the avatar.

This solution addresses the case where the VAL user is the UE subscriber.

## 6.Y Solution #Y: <Solution Name>

### 6.Y.1 Introduction

Editor’s Note: Each solution should list the key issues being addressed.

### 6.Y.2 Solution details

### 6.Y.3 Evaluation

Editor’s Note: Each solution should motivate how the potential security requirements of the key issues being addressed are fulfilled.

# 7 Conclusions

Editor’s Note: This clause contains the agreed conclusions that will form the basis for any normative work.

## 7.1 Conclusion on Key Issue #1

For spatial localization service authorization SEAL service authorization procedure to validate the VAL user to access the SEAL services is used as specified in TS 33.434 [4].

When CAPIF is used as specified in TS 23.434 [7], the security mechanism for CAPIF is used as specified in TS 33.122 [5].

## 7.2 Conclusion on Key Issue #2

When CAPIF is used, the user information is only exposed if CCF obtains permission from the resource owner as specified in TS 33.122 [5].

NOTE 1: whether CAPIF RNAA supports the case where the resource owner is different from the user of the application invoker is to be decided in normative phase, depending on the progress on CAPIF\_Ph3-Sec.

NOTE 2: whether CAPIF RNAA supports service operation level and resource level granularity in RNAA is to be decided in normative phase, depending on the progress on CAPIF\_Ph3-Sec.

Editor’s Note: Further conclusion on the reuse of user consent framework defined in TS 33.501 [6] Annex V for user privacy is FFS.

Editor’s Note: when CAPIF is not used, whether privacy protection for exposure of user sensitive information in localized mobile metaverse services through the SEAL layer can be supported by the SEAL security framework is FFS.

## 7.3 Conclusion on Key Issue #3

The TS 23.438 [8] specifies in sufficient detail the requirements, architecture, procedures and APIs for digital asset management, the digital asset service is part of the SEAL services specified in 3GPP TS 23.434 [7].

The normative work of the KI#3 aligns with the architecture of digital asset management defined in the TS 23.438 [8] and the below security solution principles should be followed:

* Security for the SEAL interfaces, especially SEAL-S, SEAL-UU, specified in TS 33.434 [4] is applied for protection of digital asset management interfaces such as DA-S, DA-UU as specified in TS 23.438 [8].
* When CAPIF is used as specified in TS 23.434 [7], the security mechanism for CAPIF specified in TS 33.122 [5] shall be followed for authenticating and authorizing a digital asset requestor (API Invoker) to access the digital asset services.

NOTE 1: Authorization of DA requestor (API Invoker) at operation and resource level is to be decided in normative phase, depending on the progress on CAPIF\_Ph3-Sec.

NOTE 2: The assumption is that the VAL user behind the DA requestor is the UE subscriber.

* When CAPIF is not used, SEAL identity management service can be utilized to authorize DA requestor for DA service.
* The digital asset access (e.g. discovery, CRUD) is authorized by the DA server based on e.g. DA profile or authorization information from CCF or access token from CCF/SIM-S or combinations of them.

## 7.4 Conclusion on Key Issue #4

The authentication of a digital representation to represent a user or authorization of a user to use the digital representation in mobile metaverse services is achieved when avatar authenticity is successfully verified and the association between avatar and user is identified.

The association between avatar and user is identified based on the information of list of owners or allowed user list and allowed metaverse application list stored in the avatar profile indicated by the avatar ID.

NOTE 1: It is assumed that user ID in mobile metaverse services can refer to user ID or VAL user ID defined in TS 23.434 [7] clause 7.

When CAPIF is used as specified in TS 23.434 [7], the security mechanisms for CAPIF specified in TS 33.122 [5] are followed for authenticating and authorizing the API invoker in the VAL UE to retrieve avatar profile from the DA server via DA service API.

NOTE 2: The assumption is that the VAL user using the avatar is the UE subscriber.

When CAPIF is not used, SEAL identity management service specified in TS 33.434 [4] is utilized for authorizing the user/subscriber to use the digital representation (avatar) in mobile metaverse services.

Annex <X> (informative):  
Change history

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Change history** | | | | | | | |
| **Date** | **Meeting** | **TDoc** | **CR** | **Rev** | **Cat** | **Subject/Comment** | **New version** |
| 2024-04 | SA3#115 Adhoc-e | S3-241422 |  |  |  | Skeleton for TR 33.721 | 0.0.0 |
| 2024-04 | SA3#115 Adhoc-e | S3-241632 |  |  |  | S3-241584, S3-241548, S3-241549, S3-241553, S3-241554 | 0.1.0 |
| 2024-05 | SA3#116 | S3-242608 |  |  |  | S3-242583 implemented | 0.2.0 |
| 2024-08 | SA3#117 | S3‑243720 |  |  |  | S3-243650, S3-243651, S3-243715, S3-243652, S3-243653, S3-243654, S3-243275 | 0.3.0 |
| 2024-10 | SA3#118 | S3-244304 |  |  |  | S3-244067, S3-244107, S3-244471, S3-244472, S3-244473, S3-244518, S3-244476 | 0.4.0 |
| 2024-11 | SA3#119 | S3-245204 |  |  |  | S3-245287, S3-245288, S3-245289, S3-245290, S3-245291, S3-245354, S3‑245292 | 0.5.0 |
| 2025-01 | SA3#119Adhoc-e | S3-250234 |  |  |  | S3-250146, S3-250149, S3-250178, S3-250179, S3-250180, S3-250181, S3-250190, S3-250211, S3-250213, S3-250214, S3-250215 | 0.6.0 |
| 2025-02 | SA3#120 | S3-250974 |  |  |  | S3-250890, S3-251049, S3-251050, S3-251065, S3-251147, S3-251148 | 0.7.0 |