

ETSI TS 133 434 V19.0.0 (2026-01)



TECHNICAL SPECIFICATION

**LTE;
5G;
Security aspects of Service Enabler Architecture Layer (SEAL)
for verticals
(3GPP TS 33.434 version 19.0.0 Release 19)**



Reference

RTS/TSGS-0333434vj00

Keywords

5G,LTE,SECURITY

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Important notice

The present document can be downloaded from the
[ETSI Search & Browse Standards application](#).

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format on [ETSI deliver repository](#).

Users should be aware that the present document may be revised or have its status changed,
this information is available in the [Milestones listing](#).

If you find errors in the present document, please send your comments to
the relevant service listed under [Committee Support Staff](#).

If you find a security vulnerability in the present document, please report it through our
[Coordinated Vulnerability Disclosure \(CVD\)](#) program.

Notice of disclaimer & limitation of liability

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.

No recommendation as to products and services or vendors is made or should be implied.

No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.

In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

Copyright Notification

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.

The content of the PDF version shall not be modified without the written authorization of ETSI.

The copyright and the foregoing restriction extend to reproduction in all media.

© ETSI 2026.
All rights reserved.

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: "*Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards*", which is available from the ETSI Secretariat. Latest updates are available on the [ETSI IPR online database](#).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™**, **LTE™** and **5G™** logo are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM®** and the GSM logo are trademarks registered and owned by the GSM Association.

Legal Notice

This Technical Specification (TS) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities. These shall be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between 3GPP and ETSI identities can be found at [3GPP to ETSI numbering cross-referencing](#).

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the [ETSI Drafting Rules](#) (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Contents

Intellectual Property Rights	2
Legal Notice	2
Modal verbs terminology.....	2
Foreword.....	6
1 Scope	8
2 References	8
3 Definitions of terms, symbols and abbreviations	9
3.1 Terms.....	9
3.2 Symbols.....	9
3.3 Abbreviations	10
4 SEAL security requirements	10
4.1 VAL user authentication and authorization.....	10
4.2 Inter-domain	10
5 Procedures	10
5.0 General	10
5.1 Security for the SEAL interfaces.....	11
5.1.1 Security for the application plane interfaces.....	11
5.1.1.0 General	11
5.1.1.1 SEAL-X1	11
5.1.1.2 SEAL-X2	11
5.1.1.3 IM-UU.....	11
5.1.1.4 KM-UU and KM-S	11
5.1.1.5 SEAL-UU.....	11
5.1.1.6 VAL-UU	11
5.1.1.7 SEAL-C.....	12
5.1.1.8 SEAL-S	12
5.1.1.9 SEAL-E.....	12
5.1.2 Security for the Signalling control plane interfaces.....	12
5.1.2.1 Security for HTTP interfaces	12
5.1.2.2 Security for LWP interfaces.....	12
5.1.3 Security for the network domain interfaces	12
5.1.4 Security for the network domain interfaces in EPS	13
5.2 User authentication and authorization	13
5.2.1 VAL user authentication.....	13
5.2.2 SEAL service authorization	13
5.2.3 Identity management functional model.....	13
5.2.4 Authentication framework	14
5.2.5 Authorization framework.....	15
5.2.6 VAL service authorization.....	15
5.3 SEAL key management procedure.....	15
5.3.1 General.....	15
5.3.2 SEAL KM Request message.....	16
5.3.3 SEAL KM Response message	17
5.4 Security procedures for interconnection.....	19
5.5 Authentication and authorization of devices over LWP interfaces	19
5.6 Security for inter-system switching between 5G and LTE.....	19
5.7 Security for VAL services over 5GS supporting EPS interworking.....	19
5.8 SEAL key provisioning procedure	19
5.8.1 General.....	19
5.8.2 SEAL KP Request message	20
5.8.3 SEAL KP Response message.....	21
5.9 Security for AIMLE Services.....	22

6	Security procedures for mobile metaverse services	22
6.1	Authentication and authorization for spatial localization services	22
6.1.1	Authentication and authorization for spatial anchor/map services when CAPIF is used	22
6.1.2	Authentication and authorization for spatial anchor/map services when CAPIF is not used	23
6.2	Authentication and authorization for digital asset services	23
6.2.1	Authentication and authorization for digital asset services when CAPIF is used	23
6.2.2	Authentication and authorization for digital asset services when CAPIF is not used	24
6.3	Authentication and authorization of digital representation	26
6.3.1	Authentication and authorization of digital representation when CAPIF is used	26
6.3.2	Authentication and authorization of digital representation when CAPIF is not used	27
6.3.2.1	General	27
6.3.2.2	The case where avatar is downloaded and present by the VAL client	28
6.3.2.2.1	VAL user authentication supporting avatar authentication and authorization	28
6.3.2.2.2	Authorization procedure for metaverse service access by an avatar	28
6.3.2.3	Case where avatar is downloaded and present by the application	29
6.4	Privacy protection for user information exposure	31
Annex A (normative): OpenID connect profile for VAL		32
A.1	General	32
A.2	VAL tokens	32
A.2.1	ID token	32
A.2.1.1	General	32
A.2.1.2	Standard claims	32
A.2.1.3	VAL claims	32
A.2.2	Access token	33
A.2.2.1	Introduction	33
A.2.2.2	Standard claims	33
A.2.2.3	VAL claims	33
A.3	SIM-C registration	33
A.4	Obtaining tokens	33
A.4.1	General	33
A.4.2	Native SIM-C	34
A.4.2.1	General	34
A.4.2.2	Authentication request	34
A.4.2.3	Authentication response	35
A.4.2.4	Access token request	35
A.4.2.5	Access token response	36
A.5	Refreshing an access token	36
A.5.1	General	36
A.5.2	Access token request	37
A.5.3	Access token response	37
A.6	Using the token to access VAL resource servers	37
A.7	Token validation	38
A.7.1	ID token validation	38
A.7.2	Access token validation	38
A.8	Token revocation	38
A.9	SIM-S interface security	38
Annex B (normative): Security mechanisms for LWP interfaces		39
B.1	General	39
B.2	Communication security for CoAP	39
B.3	Authentication and authorization mechanism on CoAP	39
B.3.1	General	39
B.3.2	VAL user authentication	39

B.3.3	SEAL service authorization.....	40
B.3.4	Authorization framework	40
B.3.5	VAL service authorization	41
B.3.6	Access token.....	41
B.3.6.1	Introduction.....	41
B.3.6.2	Standard claims.....	41
B.3.6.3	VAL claims.....	41
B.3.7	Obtaining access tokens	41
B.3.7.1	Access token request.....	41
B.3.7.2	Access token response	42
Annex C (informative): Change history		43
History		44

Foreword

This Technical Specification 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.

1 Scope

The present document specifies the security features and mechanisms to support the Service Enabler Architecture Layer (SEAL) in 5G. Specifically security architecture, functional model(s), security aspects of SEAL reference points (e.g. SEAL-UU, etc.), Key Management (KM) procedures, Identity Management (IdM) procedures and SEAL access authentication and authorization for supporting efficient use and deployment of vertical applications over the 3GPP systems are specified.

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 23.434: "Service Enabler Architecture Layer for Verticals (SEAL); Functional architecture and information flows".
- [3] IETF RFC 6749: "The OAuth 2.0 Authorization Framework".
- [4] IETF RFC 6750: "The OAuth 2.0 Authorization Framework: Bearer Token Usage".
- [5] OpenID Connect 1.0: "OpenID Connect Core 1.0 incorporating errata set 1", http://openid.net/specs/openid-connect-core-1_0.html.
- [6] 3GPP TS 33.310: "Network Domain Security (NDS); Authentication Framework (AF)".
- [7] 3GPP TS 23.401: "General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access".
- [8] 3GPP TS 23.501: "System Architecture for the 5G System; Stage 2".
- [9] IETF RFC 7521: "Assertion Framework for OAuth 2.0 Client Authentication and Authorization Grants".
- [10] IETF RFC 7523: "JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants".
- [11] IETF RFC 7797: "JSON Web Signature (JWS) Unencoded Payload Option".
- [12] IETF RFC 7515: "JSON Web Signature (JWS)".
- [13] IETF RFC 7662: "OAuth 2.0 Token Introspection".
- [14] 3GPP TS 33.210: "3G security; Network Domain Security (NDS); IP network layer security".
- [15] 3GPP TS 33.222: "Generic Authentication Architecture (GAA); Access to network application functions using Hypertext Transfer Protocol over Transport Layer Security (HTTPS)".
- [16] 3GPP TS 33.501: "Security architecture and procedures for 5G system".
- [17] 3GPP TS 29.122: "T8 reference point for Northbound Application Programming Interfaces (APIs)".

- [18] IETF RFC 7252: "The Constrained Application Protocol (CoAP)".
- [19] IETF RFC 9200: "Authentication and Authorization for Constrained Environments (ACE) using the OAuth 2.0 Framework (ACE-OAuth)".
- [20] IETF RFC 8152: "CBOR Object Signing and Encryption (COSE)".
- [21] IETF RFC 9202: "Datagram Transport Layer Security (DTLS) Profile for Authentication and Authorization for Constrained Environments (ACE)".
- [22] IETF RFC 9175: "CoAP: Echo, Request-Tag, and Token Processing"
- [23] IETF RFC 8613: ""Object Security for Constrained RESTful Environments (OSCORE)".
- [24] IETF RFC 9203: "OSCORE Profile of the Authentication and Authorization for Constrained Environments Framework".
- [25] IETF RFC 9430: "Extension of the ACE CoAP-DTLS Profile to TLS".
- [26] IETF RFC 8392: "CBOR Web Token (CWT)".
- [27] IETF RFC 8747: "Proof-of-Possession Key Semantics for CBOR Web Tokens (CWTs) ".
- [28] IETF RFC 9201: "Additional OAuth Parameters for Authentication and Authorization for Constrained Environments (ACE)".
- [29] 3GPP TS 33.122: "Security aspects of Common API Framework (CAPIF) for 3GPP northbound APIs".
- [30] 3GPP TS 23.433: "Service Enabler Architecture Layer for Verticals (SEAL);Data Delivery enabler for vertical applications".
- [31] 3GPP TS 33.401: "3GPP System Architecture Evolution (SAE); Security architecture".
- [32] 3GPP TS 33.246: "3G Security; Security of Multimedia Broadcast/Multicast Service (MBMS)".
- [33] 3GPP TS 23.482: "Functional architecture and information flows for AIML Enablement Service".
- [34] 3GPP TS 23.437: "Service Enabler Architecture Layer for Verticals (SEAL); Spatial map and Spatial anchors".
- [35] 3GPP TS 23.438: "Service Enabler Architecture Layer for Verticals (SEAL); Digital Assets".

3 Definitions of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in 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 TR 21.905 [1].

For the purposes of the present document, the terms and definitions given in TS 23.434 [2] apply.

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the abbreviations given in 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 TR 21.905 [1].

SEAL	Service Enabler Architecture Layer for Verticals
SIM-C	SEAL Identity Management Client
SIM-S	SEAL Identity Management Server
SKM-C	SEAL-Key Management Client
SKM-S	SEAL Key Management Server
VAL	Vertical Application Layer
SAn	Spatial Anchor
SM	Spatial Map

4 SEAL security requirements

4.1 VAL user authentication and authorization

[SEAL-SEC-4.1-a] All users of the VAL Service shall be authenticated.

[SEAL-SEC-4.1-b] The VAL Client and the VAL Server shall mutually authenticate each other prior to providing the VAL UE with the VAL Service User profile and access to user-specific services.

[SEAL-SEC-4.1-c] The transmission of configuration data and user profile data between an authorized VAL server in the network and the VAL UE shall be confidentiality protected, integrity protected and protected from replays.

[SEAL-SEC-4.1-d] The VAL service should take measures to detect and mitigate DoS attacks to minimize the impact on the network and on VAL users.

[SEAL-SEC-4.1-e] The VAL service shall provide a means to support confidentiality of VAL user identities.

[SEAL-SEC-4.1-f] The VAL service shall provide a means to support confidentiality of VAL signalling.

4.2 Inter-domain

[SEAL-SEC-4.2-a] VAL systems should take measures to protect themselves from external attacks at the system border.

5 Procedures

5.0 General

The security procedures in this clause also applies to SEALDD as specified in TS 23.433 [30]. In the SEALDD scenario, the SEAL client, SEAL server and SEAL service are replaced by the SEALDD client, SEALDD server and SEALDD service, respectively.

5.1 Security for the SEAL interfaces

5.1.1 Security for the application plane interfaces

5.1.1.0 General

The security for the SEAL-UU, SEAL-C, SEAL-S and SEAL-E interfaces in this clause also applies to the corresponding SEALDD interfaces (i.e. SEALDD-UU, SEALDD-C, SEALDD-S and SEALDD-E) as specified in TS 23.433 [30].

5.1.1.1 SEAL-X1

As defined in TS 23.434 [2], the SEAL-X1 reference point, exists between the key management server and the group management server and uses HTTP-1 as defined in 3GPP TS 23.434 [2] for the transport and routing of security related information to the group management server. The SEAL-X1 shall be protected using HTTPS as defined in [3], [4] and [5]. The profile for TLS implementation and usage shall follow the provisions given in 3GPP TS 33.310 [6], annex E.

5.1.1.2 SEAL-X2

The SEAL-X2 reference point enables the group management server to interact with the location management server as defined in 3GPP TS 23.434 [2]. The SEAL-X2 shall be protected using HTTPS as defined in [3], [4] and [5]. The profile for TLS implementation and usage shall follow the provisions given in 3GPP TS 33.310 [6], annex E.

5.1.1.3 IM-UU

IM-UU reference point is used between the identity management client and the identity management server. The security mechanism of SEAL-UU shall also be used for IM-UU.

The security established between the identity management server and the identity management client should be end-to-end. When this is not possible, then all sensitive material transferred between the identity management server and identity management client should be end-to-end protected with a mechanism that is out of scope of this document.

5.1.1.4 KM-UU and KM-S

The KM-UU reference point is used between the Key Management Client and Key Management Server. The security mechanism of SEAL-UU shall also be used for KM-UU.

The KM-S reference point is a direct HTTP connection used between the VAL server and the key management server and shall be protected with the same mechanism used for the SEAL-S reference point.

The security established between the KM Server and the KM client should be end-to-end. When this is not possible, then all client related material transferred between the KM server and KM client should be end-to-end protected with a mechanism that is out of scope of the present document.

5.1.1.5 SEAL-UU

A SEAL client interacts with a SEAL server over the generic SEAL-UU reference point as defined in TS 23.434 [2]. This interface shall be protected using HTTPS as defined in [3], [4] and [5] when using HTTP. The profile for TLS implementation and usage shall follow the provisions given in TS 33.310 [6], annex E. When using CoAP [18], the SEAL-UU between the SEAL client and the SEAL server shall be protected as defined in [19] (e.g., DTLS, TLS or OSCORE) with the additional security enhancements specified in [22]. When (D)TLS is used with CoAP, the (D)TLS and certificate profiling shall follow TS 33.210 [14] and TS 33.310 [6]. When OSCORE is used with CoAP, the mandatory to implement provisions given by RFC 8613 [23] shall be followed.

5.1.1.6 VAL-UU

The VAL client interacts with VAL server over VAL-UU reference point as defined in TS 23.434 [2].

NOTE: Security mechanism for the VAL-UU reference point is out of scope of present document.

5.1.1.7 SEAL-C

The VAL client interacts with a SEAL client over the SEAL-C reference point as defined in TS 23.434 [2]. This reference point resides fully within the UE and therefore, security of this interface is left to the manufacturer and is out of scope for the present document.

5.1.1.8 SEAL-S

The VAL server interacts with SEAL server over SEAL-S reference point as defined in TS 23.434 [2]. The protection of this interface shall be supported according to NDS/IP as specified in TS 33.210 [14].

When CAPIF is not used, then TLS and OAuth 2.0 [3] shall be supported. When TLS is used, mutual authentication based on client and server certificates shall be performed between the SEAL server and VAL server using TLS. Certificate based authentication shall follow the profiles given in TS 33.310 [6], clause 6.1.3a. The identities in the end entity certificates shall be used for authentication and policy checks. The structure of the PKI used for the certificate is out of scope of the present document. TLS shall be used to provide integrity protection, replay protection and confidentiality protection for the interface between the SEAL server and the VAL server. Security profiles for TLS implementation and usage shall follow the provisions given in clause 6.2 of TS 33.210 [14]. After the authentication, the SEAL server determines whether the VAL server is authorized to send requests to the SEAL server. The SEAL server shall authorize the requests from VAL server using OAuth-based authorization mechanism, the specific authorization mechanisms shall follow the provisions given in RFC 6749 [3].

When CAPIF is used as specified in TS 23.434 [2], the security mechanism for CAPIF specified in TS 33.122 [29] shall be followed. CAPIF core function shall choose the appropriate CAPIF-2e security method as defined in the clause 6.5.2 in TS 33.122 [29] for mutual authentication and protection of the SEAL server – VAL server interface. Before invoking the API exposed by the SEAL server, the VAL server as API invoker shall negotiate the security method (TLS-PSK, PKI or TLS with OAuth token) with CAPIF core function and ensure the SEAL server has information to authenticate the VAL server.

5.1.1.9 SEAL-E

A SEAL server interacts with another SEAL server over SEAL-E reference point as defined in TS 23.434 [2]. The protection of this interface shall be supported according to NDS/IP as specified in TS 33.210 [14].

5.1.2 Security for the Signalling control plane interfaces

5.1.2.1 Security for HTTP interfaces

In order to authenticate the HTTP-1 reference point, authentication mechanisms shall be performed between the HTTP client and VAL UE using either certificate based authentication or pre-shared key based authentication. Certificate based authentication shall follow in annex B of 3GPP TS 33.222 [15], and the profiles given in 3GPP TS 33.310 [6]. The usage of pre-shared key based ciphersuites is specified in the TLS profile given in 3GPP TS 33.310 [6], annex E.

The HTTP-1 reference point exists between the VAL UE and the HTTP proxy. The HTTP-2 exists between the HTTP proxy and HTTP server. The HTTP-3 reference point exists between the HTTP proxies in different networks. The HTTP interfaces shall be protected using TLS. The profile for TLS implementation and usage shall follow the provisions given in 3GPP TS 33.310 [6], annex E.

5.1.2.2 Security for LWP interfaces

Security mechanisms to be used to secure the LWP interfaces depend on the realization of the interfaces. The Annex B in the present document defines security mechanism for the realizations of LWP defined in Annex C of TS 23.434 [2].

5.1.3 Security for the network domain interfaces

A VAL UE shall perform the authentication and security mechanisms as specified in 3GPP TS 33.501 [16] for 5G network access security.

To ensure security of the interfaces between network entities within a trusted domain and between trusted domains, 3GPP TS 33.210 [14] shall be applied to secure signalling messages on the reference points unless specified otherwise. SEG as specified in 3GPP TS 33.210 [14] may be used in the trusted domain to terminate the IPsec tunnel.

5.1.4 Security for the network domain interfaces in EPS

A VAL UE shall perform the authentication and security mechanisms as specified in TS 33.401 [31] for LTE network access security.

To ensure security of the interfaces between network entities within a trusted domain and between trusted domains, TS 33.210 [14] shall be applied to secure signalling messages on the reference points unless specified otherwise. SEG as specified in TS 33.210 [14] may be used in the trusted domain to terminate the IPsec tunnel.

5.2 User authentication and authorization

5.2.1 VAL user authentication

Figure 5.2.3-1 shows the Identity Management functional model which consists of the SEAL Identity Management Server (SIM-S) and SEAL Identity Management Client (SIM-C) of the UE. The IM-UU reference point between the SIM-S and SIM-C shall provide the interface for user authentication and shall support OpenID Connect 1.0 [5] and OAuth 2.0 [9, 10] when using HTTPS to obtain an access token for the VAL UE.

5.2.2 SEAL service authorization

SEAL Service Authorization procedure shall validate the VAL user to access the SEAL services. In order to gain access to SEAL services, the SEAL client shall present an access token to the SEAL server for each service of interest. If the access token is valid, then the client shall be granted to use the service.

5.2.3 Identity management functional model

The SEAL Identity Management Server (SIM-S) and the SEAL Identity Management Client (SIM-C) provide the endpoints for VAL user authentication as shown in the SEAL Identity Management functional model in figure 5.2.3-1.

The reference point IM-UU utilizes Uu reference point as described in 3GPP TS 23.401 [7] and 3GPP TS 23.501 [8]. IM-UU shall support OpenID Connect 1.0 [5] and OAuth 2.0 [9] for VAL user authentication when using HTTPS.

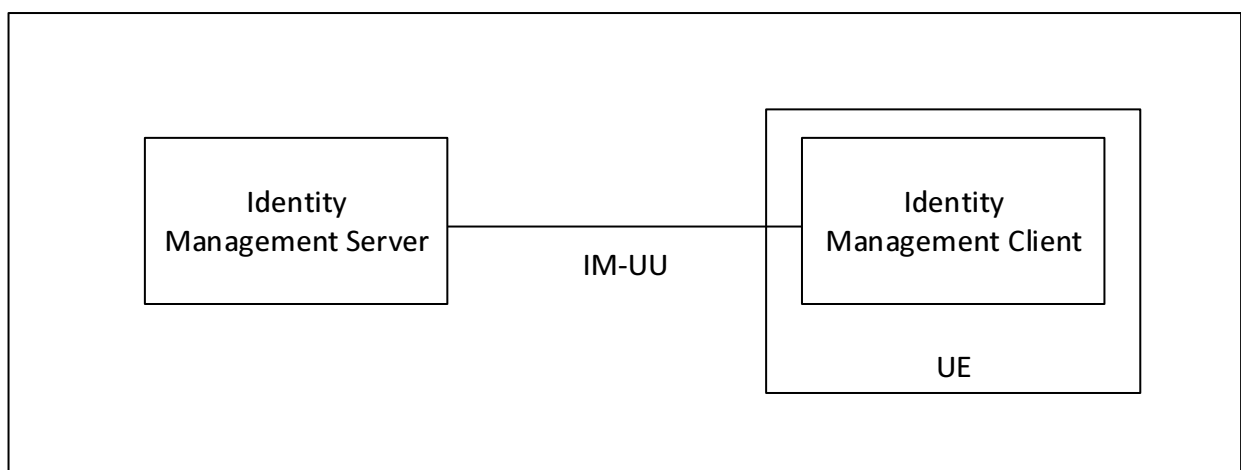


Figure 5.2.3-1: Functional model for SEAL Identity Management

In order to support VAL user authentication, the SIM-S shall be provisioned with the VAL user ID and VAL service IDs (usage of VAL user ID and VAL service ID is described in clause 7 of TS 23.434 [2]). A mapping between the VAL user ID and VAL service ID(s) shall be created and maintained in the SIM-S. When a VAL user wishes to authenticate for the VAL services, the VAL user ID and credentials are provided via the UE Identity management client to the SIM-S as per OpenID Connect 1.0 [5] when using HTTPS. The SIM-S receives and shall verify the VAL user ID

and credentials. If verification is successful, then the SIM-S returns an ID token, refresh token and access token to the UE Identity management client. The SIM-C shall learn the user's VAL service ID(s) from the ID token. Table 5.2.3-1 shows the SEAL specific tokens and their usage. These tokens are further defined in clause A.2.

Table 5.2.3-1: VAL UE authentication token

Token Type	Consumer of the Token	Description
ID token	VAL UE client(s)	Contains the VAL service ID for at least one authorized VAL service.
Access token	SKM-S, SEAL service server(s)	Short-lived token (definable in the SIM-S) that conveys the UE's identity. This token contains the VAL service ID for at least one authorized service.
Refresh token	SIM-S (Authorization Server)	Allows VAL UE to obtain a new access token without forcing user to log in again.

To support the VAL service identity functional model, the VAL service ID(s):

- Shall be provisioned into the SEAL Identity management database and mapped to VAL UE IDs.
- Shall be provisioned into the SEAL Key management server (SKM-S) and mapped to UE specific key material.

5.2.4 Authentication framework

Figure 5.2.4-1 describes the VAL Authentication Framework using the OpenID Connect protocol. when using HTTPS It describes the steps by which a VAL UE authenticates to the SIM-S, resulting in a set of credentials delivered to the UE uniquely identifying the VAL service ID(s). The authentication framework supports extensible user authentication solutions based on the VAL service provider policy (shown as step 3). User authentication methods in support of step 3 (e.g. biometrics, secureID, etc.) are possible but not defined here.

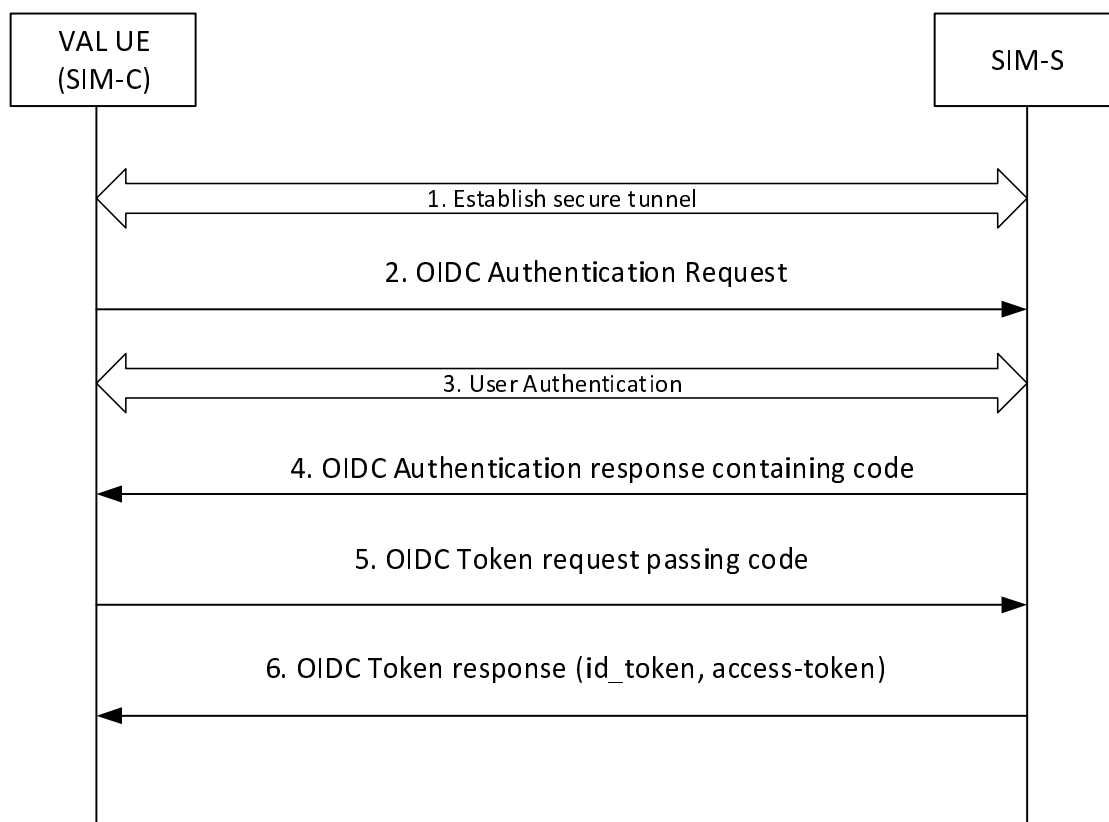


Figure 5.2.4-1: OpenID Connect (OIDC) flow supporting VAL user authentication

Step 1: VAL UE establishes a secure tunnel with the SIM-S.

Step 2: VAL UE sends an OpenID Connect Authentication Request to the SIM-S. The request may contain an indication of authentication methods supported by the UE.

Step 3: User Authentication is performed between VAL UE and the SIM-S.

NOTE: The primary credentials for user authentication (e.g. biometrics, secureID, OTP, username/password) are based on VAL service provider policy. The method chosen by the VAL service provider for authentication and authorization is neither defined nor limited by the present document, it depends on the Vertical services and authentication and authorization methods supported by it.

Step 4: SIM-S sends an OpenID Connect Authentication Response to the UE containing an authorization code.

Step 5: UE sends an OpenID Connect Token Request to the SIM-S, passing the authorization code.

Step 6: SIM-S sends an OpenID Connect Token Response to the UE containing an ID token and an access token (each which uniquely identify the user of the VAL service or key management service). The ID token is consumed by the UE to personalize the VAL client for the VAL user, and the access token is used by the UE to communicate and authorize the identity of the VAL user to the VAL server(s) and the VAL services.

5.2.5 Authorization framework

Authorization framework when using HTTP is shown in figure 5.2.5-1. A secure HTTP tunnel using HTTPS between VAL UE and VAL server shall be established before VAL service authorization. Subsequent VAL service authorization messaging make use of this tunnel. The service clients in the VAL UE present the access tokens to the VAL server over HTTP. The VAL server authorizes the user for the requested services only if the access token is valid. The procedures may be repeated as necessary to obtain additional VAL user authorizations.

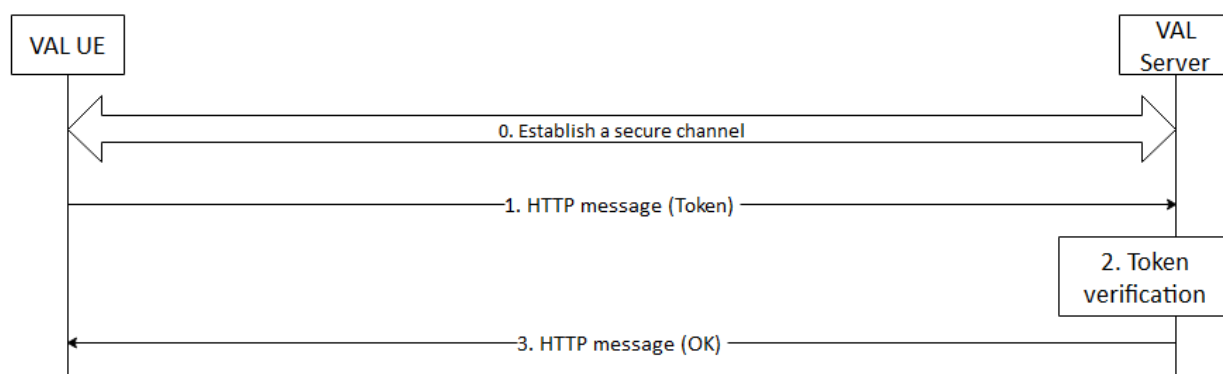


Figure 5.2.5-1: VAL User Service Authorization

After the VAL UE establishing a secure connection with the VAL server, the VAL UE sends an HTTP message containing the access token to the VAL server where service authorization is requested. The VAL server receives the message and validates the access token. If the access token is valid, The VAL server positively acknowledges the request. The VAL server may provide service related information to the VAL UE at this time.

5.2.6 VAL service authorization

The VAL service authorization procedure shall validate the VAL user authorized to access the VAL services. In order to gain access to VAL services, the VAL client shall present an access token to the VAL server for each VAL service of interest (see clause 5.2.5). If the access token is valid, then the VAL client shall be granted use of the requested VAL service.

5.3 SEAL key management procedure

5.3.1 General

To enable security for VAL services, a SEAL KM client (located in either a VAL UE or VAL server) may request key material applicable to a particular VAL service, VAL client or user.

Prior to making a key management request to the SEAL KMS (SKM-S), the VAL client or VAL user shall be authenticated by the SEAL identity management service (clause 5.2). In addition, secure connections shall be established between the SEAL client and the SKM-S (reference point KM-UU) and the VAL server and the SKM-S (reference point KM-S) prior to any associated key management requests.

As a result of the SEAL identity management authentication procedure, an access token scoped for key management services is provisioned to the SEAL UE. This access token is provided with each and every key management request to the SKM-S.

A VAL server is provisioned with an access token scoped for SEAL key management services and is provided with each and every key management request to the SKM-S. The method for provisioning this access token into the VAL server is out of scope of the present document.

Figure 5.3.1-1 shows the SEAL key management procedure. A SKM client may send a SEAL KM Request message to the SKM-S. The SKM-S validates and processes the request and responds with a SEAL KM Response message. The response contains key management material specific to the SEAL service or the VAL server request, or alternatively, an error code if the SKM-S encounters a failure condition.

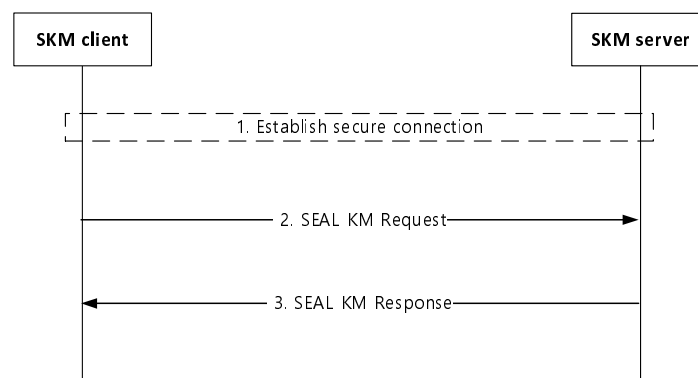


Figure 5.3.1-1: SEAL key management procedure

The procedure in figure 5.3.1-1 is described here:

1. The SKM-C establishes a secure connection, using the mechanism specified in clause 5.1.1.4, to the SKM-S. Steps 2 and 3 are within this secure connection.
2. The SKM-C sends a SEAL KM Request message to the SKM-S. The request contains the authorization credentials obtained during authentication and message content specified in clause 5.3.2.
3. The SKM-S authorizes the request and if valid, sends a SEAL KM Response message containing the requested key material (or error code) as specified in clause 5.3.3.

As a successful result of this procedure, the VAL UE or VAL Server has securely obtained service specific key material for use within the VAL system.

5.3.2 SEAL KM Request message

A SKM-C may send a SEAL KM Request message to the SKM-S. This request shall be protected (using the mechanism specified in clause 5.1.1.4) and shall contain the access token acquired during the SEAL identity management authentication procedure (clause 5.2).

The content of the SEAL KM Request is shown in table 5.3.2-1.

Table 5.3.2-1: Contents of a SEAL KM Request

Name	Description
Version	The version number of the SEAL key management request.
SKmsUri	The URI of the SKM-S to which the request is sent.
ServiceID	A string representing the VAL service/application related to the VAL client request.
ClientID	(Optional) A string representing the client. (see NOTE)
DeviceID	(Optional) A string representing the device. (see NOTE).
UserID	(Optional) A string representing the user. (see NOTE)
Date/Time	The Date and Time of the request. This number represents the number of seconds from 1970-01-01T0:0:0Z as measured in UTC.

NOTE: Only one of these fields may be present in any given SEAL KMS Request message.

The identities listed in table 5.3.2-1 map to SEAL identities defined in 3GPP TS 23.434 [2]. Namely, the ServiceID maps to the VAL service identity (VAL service ID), the ClientID maps to the VAL client or client on the VAL server, the DeviceID maps to the VAL UE identity (VAL UE ID), and the UserID maps to the VAL user identity (VAL user ID).

The 'Version' field identifies the version of the SEAL KM Request message. The current version is defined as "1.0.0".

The 'Date/Time' field is used primarily as an anti-replay mechanism for SEAL key management requests and responses. If the 'Date/Time' field is significantly out of range (more than a few seconds), this could indicate a replay attack.

Upon receipt of a SEAL KM Request message, the SKM-S shall verify that:

- the access token is valid;
- the signature is valid;
- the SKmsUri is the SKM-S URI of the target SEAL KMS where the key information is stored; and
- the Date/Time is within a recent time window (e.g. 5 seconds).

If valid, the request is accepted and processed by the SKM-S. A standalone ServiceID, or a ServiceID in combination with a ClientID, DeviceID, or UserID may be present in the SEAL KM Request message. This combination may be used by the KMS to identify a specific key material record. Each key management record may be unique to a VAL application or VAL service. The format and content of a key management record is defined and securely provisioned into the SEAL KMS by the VAL application or VAL service owner/operator.

A SEAL KM client (SKM-C) located in the VAL server may use the SEAL key provisioning procedure described in clause 5.8 to provision the VAL service or VAL application key material into the KMS.

The method used to organize, manage, and maintain VAL service or VAL application key material within the KMS is out of scope of the present document.

5.3.3 SEAL KM Response message

The SEAL KM Response message is sent to the SKM-C in response to a SEAL KM Request message.

A successful SEAL key management procedure results in a SEAL KM Response message, which typically includes a payload containing key management information uniquely applicable to the requested service, client or user. If an error occurs, an error code may be returned in the SEAL KM Response message.

The SEAL KM Response message shall be protected in transit using the mechanism specified in clause 5.1.1.4. The Payload within a SEAL KM Response message may be protected end-to-end between the SKM-C and SKM-S depending on the applicability of the underlying VAL service making the request. The method for securing a Payload end-to-end between the SKM-C and the SKM-S is outside the scope of the present document. The key material contents provided in a Payload are defined by the underlying VAL service and are outside the scope of the present document.

The content of a SEAL KM Response message is shown in table 5.3.3-1.

Table 5.3.3-1: Contents of a SEAL KM Response message

Name	Description
UserUri	URI of the user for which the response is intended.
SKmsUri	The URI of the SKM-S sending the response.
ServiceID	A string representing the VAL service/application related to the VAL client request. This is the same field as received in the SEAL KM Request message.
SKmsID	(Optional) The ID of the SKM-S providing the response message.
ClientID	(Optional) A string representing the client (see NOTE)
DeviceID	(Optional) A string representing the device (see NOTE)
UserID	(Optional) A string representing the user. (see NOTE)
Date/Time	The Date and Time of the response. This number represents the number of seconds from 1970-01-01T0:0:0Z as measured in UTC.
ErrorCode	(Optional) Reason code indicating the failure of the requested action. If not present, the key management request is assumed to be successful.
Payload	(Optional) Key management payload specific to the VAL user, client or application. This field is not be present if an error occurs .
NOTE: If this field is present in the SEAL KM Request message then this field shall be present in the SEAL KM Response message and shall be the same value.	

The identities listed in table 5.3.3-1 are described in clause 5.3.2.

If the SKM-S does not encounter an error during processing of the SEAL KM Request message, the SEAL KM Response message carries a set of security parameters contained in the "Payload" field.

If the SKM-S encounters an error while processing the SEAL KM Request message, an error value described in table 5.3.3-2 shall be returned in the 'ErrorCode' field of the SEAL KM Response message and the 'Payload' field shall not be present.

In the event of an error, the user and/or the operator of the VAL service, UE, or client may be notified.

Table 5.3.3-2: 'ErrorCode' values

ErrorCode	Description	Maps To
01	Unspecified error	"500 Internal Server Error" as described in Table 5.2.6-1 of TS 29.122 [17]
02	Key Information not available for specified service, client, device or user.	"404 Not Found" as described in Table 5.2.6-1 of TS 29.122 [17]
03	Request rejected	"401 Unauthorized" as described in Table 5.2.6-1 of TS 29.122 [17]
04	Unable to validate request	"400 Bad Request" or "403 Forbidden" as described in Table 5.2.6-1 of TS 29.122 [17]
05-FF	Reserved	N/A

The selection of the key material returned in the Payload of a SEAL KM Response message is determined by the ServiceID and (optionally) the ClientID, DeviceID or UserID. The combination of the ServiceID with the ClientID, DeviceID or UserID allows the VAL service to request a more specific set of key material.

For example, if a ClientID is included in the SEAL KM Request message, the KMS may return a Payload that contains a set of client specific key material applicable to the ClientID within the requesting VAL service (ServiceID). If the DeviceID is included, the KMS may return a Payload that contains device specific key material applicable to the DeviceID within the requesting VAL service (ServiceID). If the UserID is included, the KMS may return a Payload that contains user specific key material applicable to that UserID within the requesting VAL service (ServiceID).

5.4 Security procedures for interconnection

Interconnection between a primary VAL system and a partner VAL system is specified in 3GPP TS 23.434 [2].

A VAL client shall perform user authorization only to VAL servers within their own VAL system. When communication is required by a VAL client from another interconnected VAL system, user authorization takes place in the serving VAL system and follows the VAL user service authorization procedures as defined in clause 5.2.

VAL systems should protect themselves at the system border from external attackers.

5.5 Authentication and authorization of devices over LWP interfaces

Authentication and authorization mechanism for devices over LWP interfaces depends on the application protocol. The Annex B in the present document defines authentication and authorization procedures for the realizations of application protocols defined in Annex C of TS 23.434 [2].

5.6 Security for inter-system switching between 5G and LTE

During inter-system mobility from 5G MBS session to LTE eMBMS/unicast bearer or from LTE eMBMS to 5G MBS sessions (either broadcast or multicast), when the target system is EPS, the security protection specified in TS 33.246 [32] applies and when the target system is 5GS, the security protection specified in TS 33.501 [16] applies.

5.7 Security for VAL services over 5GS supporting EPS interworking

The VAL server consumes the network resource management services from the NRM server. For the VAL services over 5GS supporting EPS interworking, the security mechanisms as specified in TS 33.501 [16] are followed.

5.8 SEAL key provisioning procedure

5.8.1 General

The SEAL key provisioning procedure may be used by a SEAL KM client (SKM-C) located in a VAL server to provision key information applicable to a particular VAL service, VAL client, VAL device, or VAL user.

A VAL server shall be provisioned with an access token scoped for SEAL key provisioning services. The method for provisioning this access token into the VAL server is out of scope of the present document. The VAL server using the SKM-C shall provide this access token with every key provisioning request made to the SKM-S. In addition, a secure connection shall be established between the SKM-C and the SKM-S prior to any associated key provisioning requests.

The KMS shall authenticate and validate the presented access token (i.e. verifying that the SKeyProv parameter defined in clause A.2.2.3 is provided and correct), and shall validate that the requesting SKM-C has the authorization to perform key provisioning.

Figure 5.8.1-1 shows the SEAL key provisioning procedure. A SKM-C may send a SEAL Key Provisioning Request message to the SKM-S. The SKM-S shall validate and process the request and respond with a SEAL KP Response message. The request contains key information specific to a particular VAL service, VAL client, VAL device, or VAL user. The SEAL KP Response message provides either an acknowledgement or an error code (if the SKM-S encounters a failure condition).

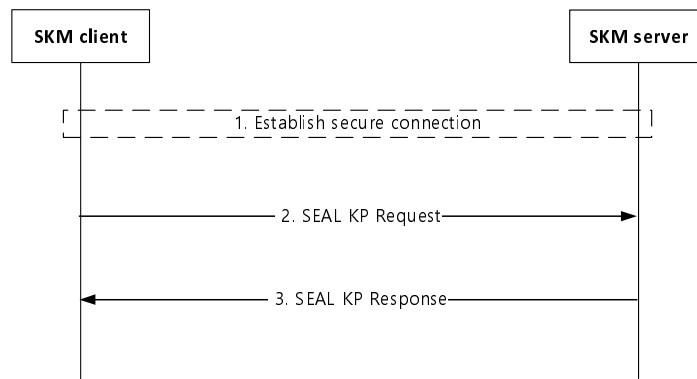


Figure 5.8.1-1: SEAL key provisioning procedure

The procedure in figure 5.8.1-1 is described here:

1. The SKM-C establishes a secure connection to the SKM-S using the mechanism specified in clause 5.1.1.4. Steps 2 and 3 are within this secure connection.
2. The SKM-C sends a SEAL KP Request message to the SKM-S. The request contains the authorization credentials (i.e. access token) and message content specified in clause 5.8.2.
3. The SKM-S validates the credentials and verifies that the requesting SKM-C is an authorized key provisioning client. Upon authorization, the SKM-S processes the request and returns a SEAL KP Response message to the SKM-C containing an acknowledgement (or error code) as specified in clause 5.8.3.

As a successful result of this procedure, the VAL Server has securely provisioned specific key information for use within the VAL system for a particular VAL service, VAL device, VAL client, or VAL user.

5.8.2 SEAL KP Request message

An authorized SKM-C may send a SEAL KP Request message to the SKM-S. This request shall be protected (using the mechanism specified in clause 5.1.1.4) and shall contain the access token scoped for SEAL key provisioning (clause A.2).

The content of the SEAL KM Request is shown in table 5.8.2-1.

Table 5.8.2-1: Contents of a SEAL KP Request

Name	Description
Version	The version number of the SEAL key provisioning request.
SValClientUri	The URI of the SKM-C (hosted in the VAL server) making the request.
SKmsUri	The URI of the SKM-S to which the request is sent.
ServiceID	A string representing the VAL service/application related to the VAL client request.
ClientID	(Optional) A string representing the client related to the key material in the Payload. (see NOTE)
DeviceID	(Optional) A string representing the device related to the key material in the Payload. (see NOTE)
UserID	(Optional) A string representing the user related to the key material in the Payload. (see NOTE)
Date/Time	The Date and Time of the request. This number represents the number of seconds from 1970-01-01T0:0:0Z as measured in UTC.
KP PayloadID	(Optional) A string identifier representing the Payload.
KP Payload	Key provisioning payload specific to the identified VAL ServiceID, UserID, ClientID, or DeviceID.
NOTE: Only one of these fields may be present in any given SEAL KP Request message.	

The identities listed in table 5.8.2-1 map to SEAL identities defined in 3GPP TS 23.434 [2], and identify the endpoints targets of the key information (Payload). Namely, the ServiceID maps to the VAL service identity (VAL service ID), the ClientID maps to the VAL client or client on the VAL server, the DeviceID maps to the VAL UE identity (VAL UE ID), and the UserID maps to the VAL user identity (VAL user ID).

The 'Version' field identifies the version of the SEAL KP Request message. The current version is defined as "1.0.0".

The 'Date/Time' field is primarily as an anti-replay mechanism for SEAL key provisioning requests and responses. If the 'Date/Time' field is significantly out of range (more than a few seconds), this could indicate a replay attack.

Upon receipt of a SEAL KP Request message, the SKM-S shall verify that:

- the access token is valid and contains the SKeyProv field;
- the signature is valid;
- the requesting SKM-C is authorized for key provisioning;
- the SKmsUri is the SKM-S URI of the target SEAL KMS where the key information shall be stored; and
- the Date/Time is within a recent time window (e.g. 5 seconds).

If valid, the request is accepted and processed by the SKM-S. A standalone ServiceID, or a ServiceID in combination with a ClientID, DeviceID, or UserID may be present in the SEAL KP Request message. This combination may be used by the KMS to map the key material in the Payload with a specific client, device, or user. The format and content of a key provisioning Payload is defined by the VAL application or VAL service owner/operator and is out of scope of this document. For example, such content may include VAL specific keys, additional tokens, credentials, or other important security related information.

The method used to organize, manage, and maintain VAL service or VAL application key material within the KMS is out of scope of the present document.

5.8.3 SEAL KP Response message

The SEAL KP Response message is sent by the SKM-S to the SKM-C in response to a SEAL KP Request message.

A successful SEAL key provisioning procedure results in the KMS sending a SEAL KP Response message containing an acknowledgement indicating the KMS successfully received and processed the SEAL KP Request message. If the KMS is unable to successfully process the SEAL KP Request message, the KMS may instead return an error code in the SEAL KP Response message.

The SEAL KP Response message shall be protected in transit using the mechanism specified in clause 5.1.1.4.

The content of a SEAL KP Response message is shown in table 5.8.3-1.

Table 5.8.3-1: Contents of a SEAL KP Response message

Name	Description
SValKmcUri	URI of the VAL SKM-C client for which the response is intended.
SKmsUri	The URI of the SKM-S sending the response.
ServiceID	A string representing the VAL service/application related to the VAL client request. This is the same field as received in the SEAL KM Request message.
SKmsID	(Optional) The ID of the SKM-S providing the response message.
ClientID	(Optional) A string representing the client (see NOTE)
DeviceID	(Optional) A string representing the device (see NOTE)
UserID	(Optional) A string representing the user. (see NOTE)
Date/Time	The Date and Time of the response. This number represents the number of seconds from 1970-01-01T0:0:0Z as measured in UTC.
KP PayloadID	(Optional) A string representing the received Payload. (see NOTE)
ErrorCode	(Optional) Reason code indicating the failure of the requested action. If this field is not present, the key provisioning request is assumed to be successful.
NOTE:	If this field is present in the SEAL KP Request message then this field shall be present in the SEAL KP Response message and shall be the same value.

The identities listed in table 5.8.3-1 are described in clause 5.8.2.

If the SKM-S encounters an error while processing the SEAL KP Request message, an error value described in table 5.8.3-2 should be returned in the 'ErrorCode' field of the SEAL KP Response message.

In the event of an error, the user and/or the operator of the VAL service may be notified.

Table 5.8.3-2: 'ErrorCode' values

ErrorCode	Description	Maps To
01	Unspecified error	"500 Internal Server Error" as described in Table 5.2.6-1 of TS 29.122 [17]
02	Referenced client, device, user, or service not found.	"404 Not Found" as described in Table 5.2.6-1 of TS 29.122 [17]
03	Request rejected	"401 Unauthorized" as described in Table 5.2.6-1 of TS 29.122 [17]
04	Unable to validate request	"400 Bad Request" or "403 Forbidden" as described in Table 5.2.6-1 of TS 29.122 [17]
05-FF	Reserved	N/A

5.9 Security for AIMLE Services

The AIML Enablement (AIMLE) SEAL services are specified in TS 23.482 [33] and the AIMLE security shall follow the SEAL security aspects specified in this present document.

6 Security procedures for mobile metaverse services

6.1 Authentication and authorization for spatial localization services

6.1.1 Authentication and authorization for spatial anchor/map services when CAPIF is used

When CAPIF is used as specified in TS 23.434 [2], the security mechanism for CAPIF specified in TS 33.122 [29] is referred to protect interfaces between spatial anchors (SAn) client and spatial anchors server specified in TS 23.437 [34], between VAL server and spatial anchors server, between spatial map (SM) client and spatial map server specified in TS 23.437 [34], VAL server and spatial map server. Spatial anchors client, spatial map client and VAL server take role of API Invoker in CAPIF. Spatial anchors server and spatial map server take role of AEF in CAPIF.

The authentication and authorization mechanisms specified for CAPIF-2e reference point in TS 33.122 [29] clause 6.5.2 shall be followed for authenticating and authorizing a SAn/SM client to access the spatial anchor/map services defined in TS 23.437 [34] clauses 8 and 9.

If the owner of the spatial anchor or spatial map is provided in the spatial anchor/map service request, authentication and authorization for RNAA specified in clause 6.5.3 of TS 33.122 [29] are followed to authenticate and authorize a SAn/SM client to access the spatial anchor/map of a specific owner.

NOTE 1: When applying authentication and authorization for RNAA, the owner of the spatial anchor/map has to be a UE subscriber.

NOTE 2: How CAPIF RNAA supports service operation level and resource level authorization using access token is specified in TS 33.122 [29].

6.1.2 Authentication and authorization for spatial anchor/map services when CAPIF is not used

Security for the SEAL-C, SEAL-UU and SEAL-S interfaces as specified in clause 5.1.1 is applied for protection of SAn/SM-C, SAn/SM-UU and SAn/SM-S interfaces defined in TS 23.437 [34] respectively.

The SEAL Identity Management service specified in clause 5.2.4 is utilized to authenticate the spatial anchors (SAn)/spatial map (SM) client. In support for spatial localization authorization, the access token issued by the SIM server and provisioned to the SAn/SM client during the authentication is used to gain services for SAn client or SM client.

In order to gain access to SEAL spatial anchor/map services, the SAn/SM client shall present an access token to the SAn/SM server for each spatial localization service of interest. The SAn/SM server performs authorization based on the access token by verifying the SIM-S signature using the SIM-S' certificate. If the access token is valid, then the SAn/SM client shall be granted to use the service.

Editor's Note: The clarification on how existing procedures can be utilized is FFS.

6.2 Authentication and authorization for digital asset services

6.2.1 Authentication and authorization for digital asset services when CAPIF is used

When CAPIF is used as specified in TS 23.434 [2], the security mechanism for CAPIF specified in TS 33.122 [29] is referred to protect interfaces between digital asset client and digital asset server specified in TS 23.438 [35], between VAL server and digital asset server. The digital asset (DA) requestor (e.g. digital asset client or VAL server) takes the role of API Invoker in CAPIF. Digital asset server takes role of AEF in CAPIF.

Authentication and authorization for RNAA specified in clause 6.5.3 of TS 33.122 [29] shall be followed for the DA server to authorize the DA requestor to access the DA service API allowed by the DA owner. In addition to verifying the access token, the DA server shall further check the access control list in the DA profile.

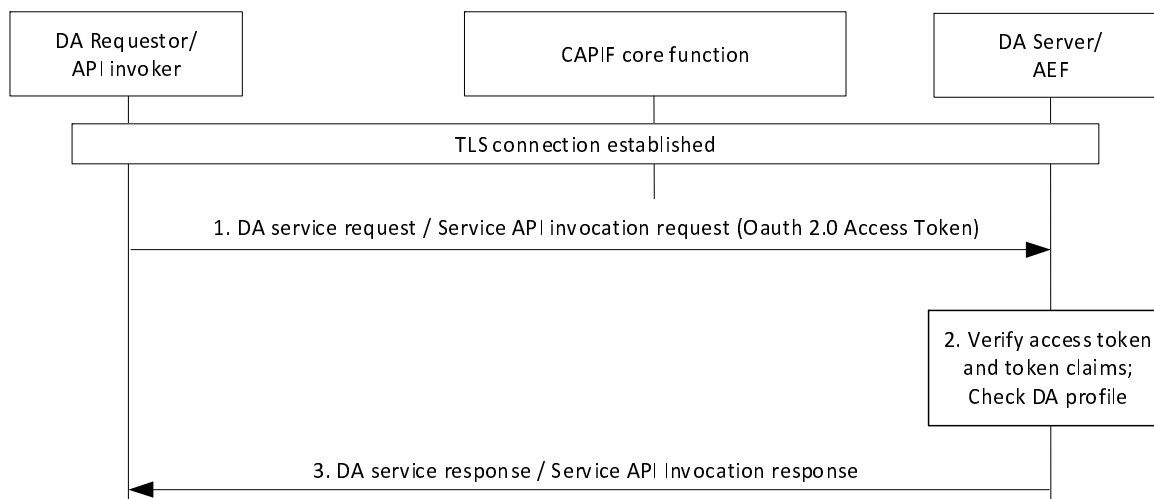


Figure 6.2.1-1: Procedure for DA requestor authorization to access DA services with token

1. The DA requestor sends the DA service request (e.g. DA create/retrieve/update/delete request) to the DA server. The access token received from the CCF is sent along with the DA service request. The DA owner identity, e.g. GPSI if the DA is owned by a UE subscriber, is included in the access token, as per the procedure in TS 33.122 [29] clause 6.5.3.
2. Upon receiving the request with an access token, the DA server validates the integrity of the token using the CCF's certificate. If validation of the token is successful, the DA server further verifies the DA service request against the authorization claims in the token.

In case the access token does not contain sufficient claims for authorizing the DA service API invocation, the DA server further checks its local DA profile for finer level authorization if needed. For example, if the DA profile contains the information of owner(s) or allowed operation(s) for resource level authorization, the DA server checks if the DA owner in the access token matches the owner(s) in the DA profile, and if the service operation in the access token matches the allowed operation(s) in the DA profile.

3. If the DA service API invocation is successfully authorized by the DA server, it executes the service logic for the invoked DA service API and returns the DA service response as a result of the DA service API invocation.

NOTE 1: The assumption is that the user behind the digital asset client is the owner of the digital asset.

NOTE 2: How CAPIF RNAA supports service operation level and resource level authorization is specified in TS 33.122 [29].

6.2.2 Authentication and authorization for digital asset services when CAPIF is not used

When CAPIF is not used, security for the SEAL interfaces, especially SEAL-S, SEAL-UU, specified in clause 5.1.1 is applied for protection of digital asset management interfaces such as DA-S, DA-UU as specified in TS 23.438 [35].

SEAL identity management service is used to support authentication of VAL user and authorization for DA services. Prior to making a DA service request to the DA server, the VAL user behind the DA client shall be authenticated using the SEAL identity management service as in clause 5.2.

In order to gain access to digital asset services, the digital asset client shall present an access token to the digital asset server for digital asset service of interest. If the access token is valid, then the digital asset client is granted to use the service.

The procedure to authenticate a VAL user and authorize the digital asset client to access digital asset is shown in figure 6.2.2-1.

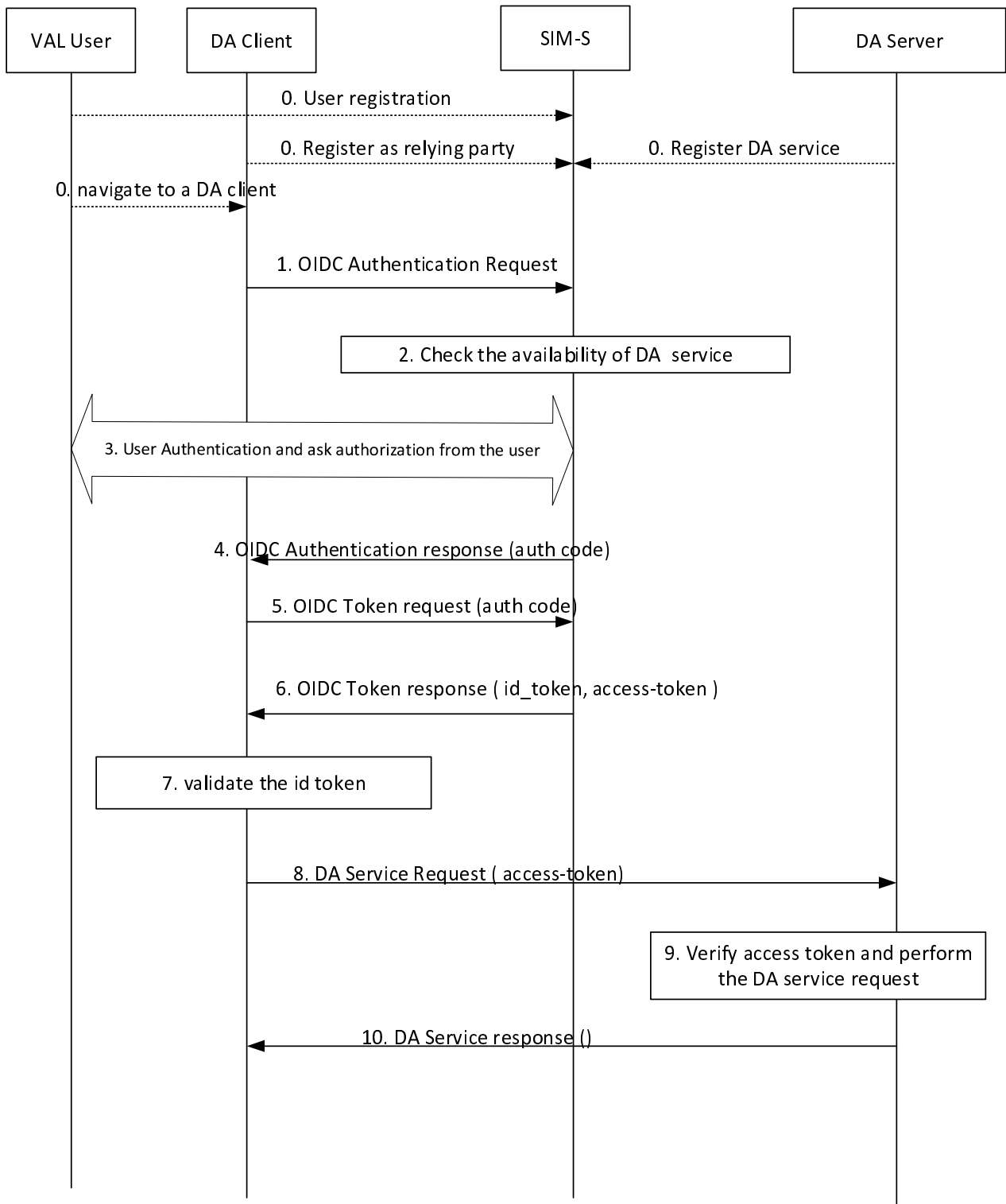


Figure 6.2.2-1. Procedure to authorize the digital asset client to access digital asset

0. DA client registers to SIM-S as Relying Party and delegates identity management and user authentication to the SIM-S, which takes the role of OpenID Provider. The DA service is registered to the SIM-S either by DA server or O&M. The VAL User registers to the SIM-S with user profile and gets User ID from the SIM-S. The VAL User navigates to a DA client with the User ID.

NOTE 1: The step 0 is implementation dependent and out of 3GPP scope.

1. With the User ID, the DA client sends OIDC authentication request to the SIM-S, including User ID, and response type. The Scope is set to open id and DA service, the response type is set to auth code.

2. The SIM-S checks if the DA service is available. In addition, the SIM-S may check if the DA client is allowed to access the DA service according to local policy.
3. The SIM-S triggers user authentication towards the VAL User and gets authorization from the VAL User to allow the DA client to access the DA service.
4. The SIM-S sends OIDC response to the DA client with an auth code.
5. The DA client sends token request to the SIM-S with auth code.
6. The SIM-S sends token response to the DA client. The response includes an ID token and an Access Token. The ID token includes issuer set to the SIM-S, audience set to the DA client, subject set to the User ID and expiration time. The Access Token includes issuer set to the SIM-S, audience set to the DA server, subject set to the User ID or VAL User ID, scope set to DA service and expiration time.

NOTE 2: The expiration time in the access token should be shorter than the one in the ID Token. I.e., the client should not be able to use the access token after the user has log-out.

7. The DA client validates the ID token to authenticate the VAL User.

Prior to any associated DA service requests, secure connections shall be established between the DA client and the DA server (reference point DA-UU).

8. The DA client sends DA service request to the DA server, including the Access Token obtained in step 6.
9. Upon receiving the request with an access token, the DA server verifies the Access Token and retrieve User ID from the subject claim of the Access Token. DA server further checks if the User ID matches one of the owners or one of the uses in the DA profile of the accessed DA. If does, the DA server further checks if the requested operation is allowed by this user based on access control list of the DA profile of the accessed DA. If does, DA server perform requested operation on the accessed DA. Otherwise, DA server rejects the DA service request.

NOTE 3: Access Token verification includes verifying digital signature of the Access Token, checking if issuer is the SIM-S, audience is the DA server, client id matches the DA client, scope is DA service, and the token is not expired.

10. After successful authorization by the DA server, the DA server sends response to the DA client, which may include the DA profile or DA media.

6.3 Authentication and authorization of digital representation

6.3.1 Authentication and authorization of digital representation when CAPIF is used

When CAPIF is used as specified in TS 23.434 [2], the metaverse application client in VAL UE shall act as the API invoker and the DA server shall act as the AEF.

As the digital asset profile and media are created in the DA server according to TS 23.438 [y], the created avatar shall be digitally signed by the DA server to ensure the authenticity of the avatar.

For authenticating the avatar and authorizing its usage by the user, the API invoker in the VAL UE shall invoke the DA service API for retrieving DA profile from the AEF (i.e. DA server). Based on the validation of avatar authenticity and the retrieved DA profile containing the association between avatar and user, the VAL UE determines whether the avatar can represent the user in the metaverse application and whether the user is allowed to use the avatar in the application. The detailed procedure is depicted in Figure 6.XY.1-1.

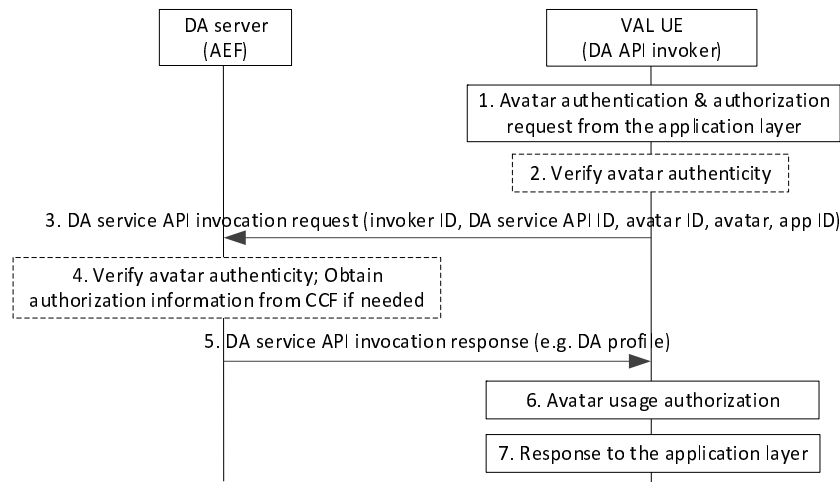


Figure 6.3.1-1: Avatar authentication and usage authorization

1. The metaverse application in the VAL UE triggers for avatar authentication and authorization by passing at least the avatar ID and corresponding avatar (media), VAL user ID, app ID.
2. If the certificate of DA server is locally available in the VAL UE, it verifies the avatar authenticity by verifying the digital signature of the avatar signed by the DA server. Only when the avatar is verified as authentic, the VAL UE then proceeds with the following steps.
3. The API invoker in the VAL UE sends a DA service API invocation request to the DA server, including at least the invoker ID, identifier of DA service API (i.e. SS_DAProfileManagement_Retrieve in TS 23.438 [y]), avatar ID and optionally application ID. If the VAL UE is not able to verify the avatar authenticity (e.g. no DA server certificate) in step #2, it also sends the received avatar to the DA server for verification.
4. If avatar is included in the request, the DA server first verifies the avatar authenticity, and then checks the application ID if received. Only when the avatar is verified as authentic and allowed to be used in the application, the DA server then accepts 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.
5. The DA server sends the DA service API invocation response to the VAL UE, which contains the requested DA profile.
6. If the requested DA profile is received, it implies that the avatar is successfully 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 the avatar and VAL user in the DA profile.

NOTE: The VAL user is assumed to be a UE subscriber.

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

6.3.2 Authentication and authorization of digital representation when CAPIF is not used

6.3.2.1 General

When CAPIF is not used, SEAL identity management service defined in clause 5.2 is used to authenticate the VAL user using an avatar to represent him/her in a metaverse service.

There are two cases an avatar is used in metaverse applications. One case is that the VAL client downloads an avatar before requesting metaverse services, which can be used in one or more metaverse applications. Then the VAL client requests metaverse services using the downloaded avatar to represent him/her. The other case is that, the metaverse application triggers to download from the DA server an authentic avatar to be used to represent the VAL user navigated to the metaverse application.

6.3.2.2 The case where avatar is downloaded and present by the VAL client

6.3.2.2.1 VAL user authentication supporting avatar authentication and authorization

During the SEAL identity management authentication procedure, the avatar authenticity shall be validated by the SIM server. As a result of the SEAL identity management authentication procedure, an access token is issued by the SIM server based on the information from the DA server.

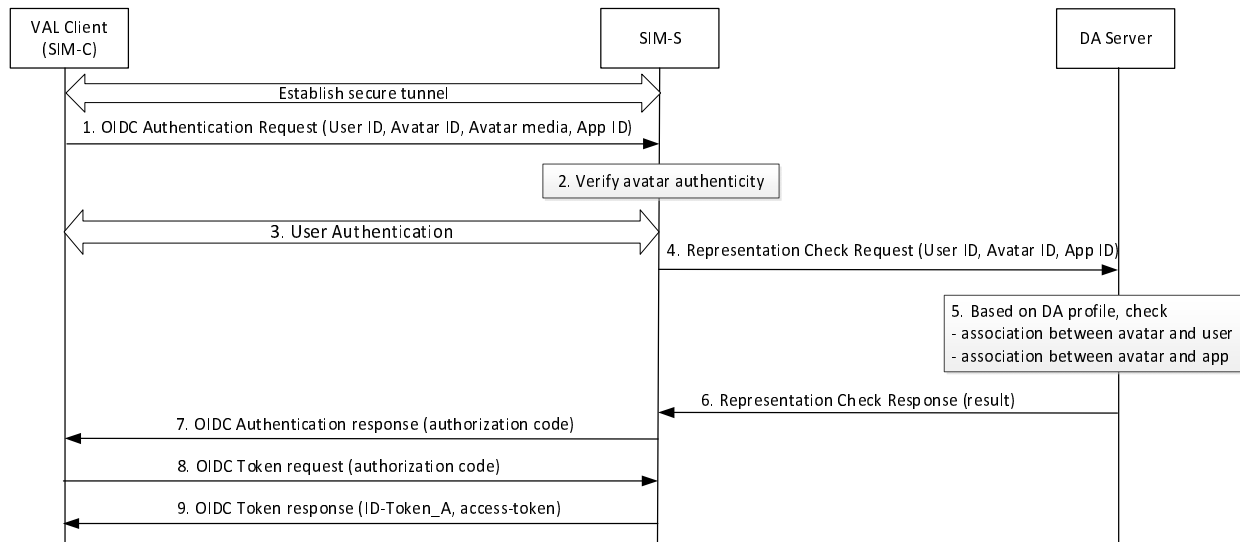


Figure 6.3.2.2.1-1: VAL user authentication supporting avatar authentication and authorization

1. When the user on the VAL UE starts to request a metaverse service using an avatar to represent him/her, the VAL client sends to the SIM-S an OpenID Connect Authentication Request, which contains the VAL client ID, user ID, avatar ID, the signed avatar media, and application ID.
2. The SIM-S verifies the avatar authenticity using the DA server's certificate.
3. If avatar authenticity is validated, step #3 in clause 5.2.4 is performed. Otherwise, an OpenID Connect Authentication Response is returned to the VAL client, indicating the avatar is unauthentic.
4. If the VAL user is authenticated, the SIM-S sends a Representation Check Request to the DA server, which contains the user ID, avatar ID and application ID.
5. Based on the received user ID, avatar ID and application ID, the DA server determines whether the avatar is allowed to be used by the user and whether the avatar is allowed to be used in the application, by checking against the DA profile.
6. The DA server returns the Representation Check Response with the check result.
7. If the check result is positive, step #4 in clause 5.2.4 is performed. Otherwise, the OpenID Connect Authentication Response indicates the mismatch between the avatar and user or between the avatar and application.
- 8-9. The same steps as steps #5 & #6 in clause 5.2.4. The access token is issued by the SIM-S based on the check result from the DA server. The details of access token refer to step #1 in clause 6.3.b.2.2.

6.3.2.2.2 Authorization procedure for metaverse service access by an avatar

The access token provisioned to the VAL client is included in the metaverse service request to the metaverse VAL server, which checks the avatar authenticity and its usage by the user based on the token.

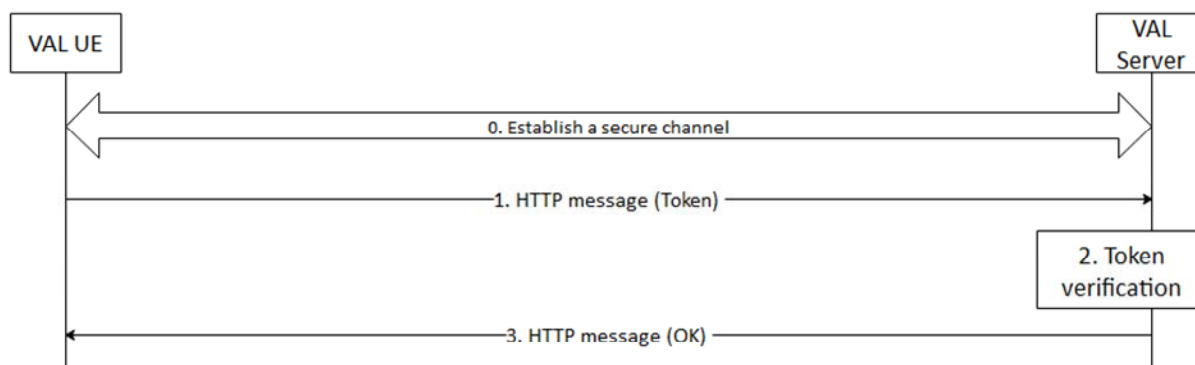


Figure 6.3.2.2-1: Authorization procedure for metaverse service access by an avatar

1. The VAL UE sends an HTTP message to the VAL server containing the user ID, avatar ID, avatar media, application ID, and access token. The access token contains the standard claims as defined in A.2.2.2, as well as additional VAL claims, such as a claim for the association between avatar and user and a claim for the signed avatar media. The standard scope claim contains the application information indicated by the application ID.
2. The VAL server validates the access token as follows:
 - check the integrity of the token using the SIM-S' certificate;
 - check whether the application ID in the request matches the information in the standard scope claim;
 - check whether the user ID and avatar ID in the request matches the VAL claim for the association between avatar and user;
 - check whether the avatar media in the request matches the VAL claim for avatar media.

If the access token is successfully verified, the VAL server determines that the avatar present by the VAL client is authentic and allowed to be used by the user in the requested metaverse service.

NOTE: When the avatar ID is not integrated into the avatar media, the avatar media needs to be included in the access token.

3. The VAL server responds the request with OK/failure or with the service information requested by the VAL client.

6.3.2.3 Case where avatar is downloaded and present by the application

A VAL User is authorized to use an avatar in a mobile metaverse service if the User ID is matched with one of list of owners in the DA profile of the corresponding avatar, and the association between the avatar and the VAL User is identified by the metaverse application based on the ID token got from SIM-S after user authentication.

The procedure to authorize a VAL user to use the avatar in metaverse service and identify the association between the avatar and the VAL User by the metaverse application is shown in figure 6.3.2.3-1.

The metaverse application plays the role of SEAL Client, and DA Server plays the role of SEAL server.

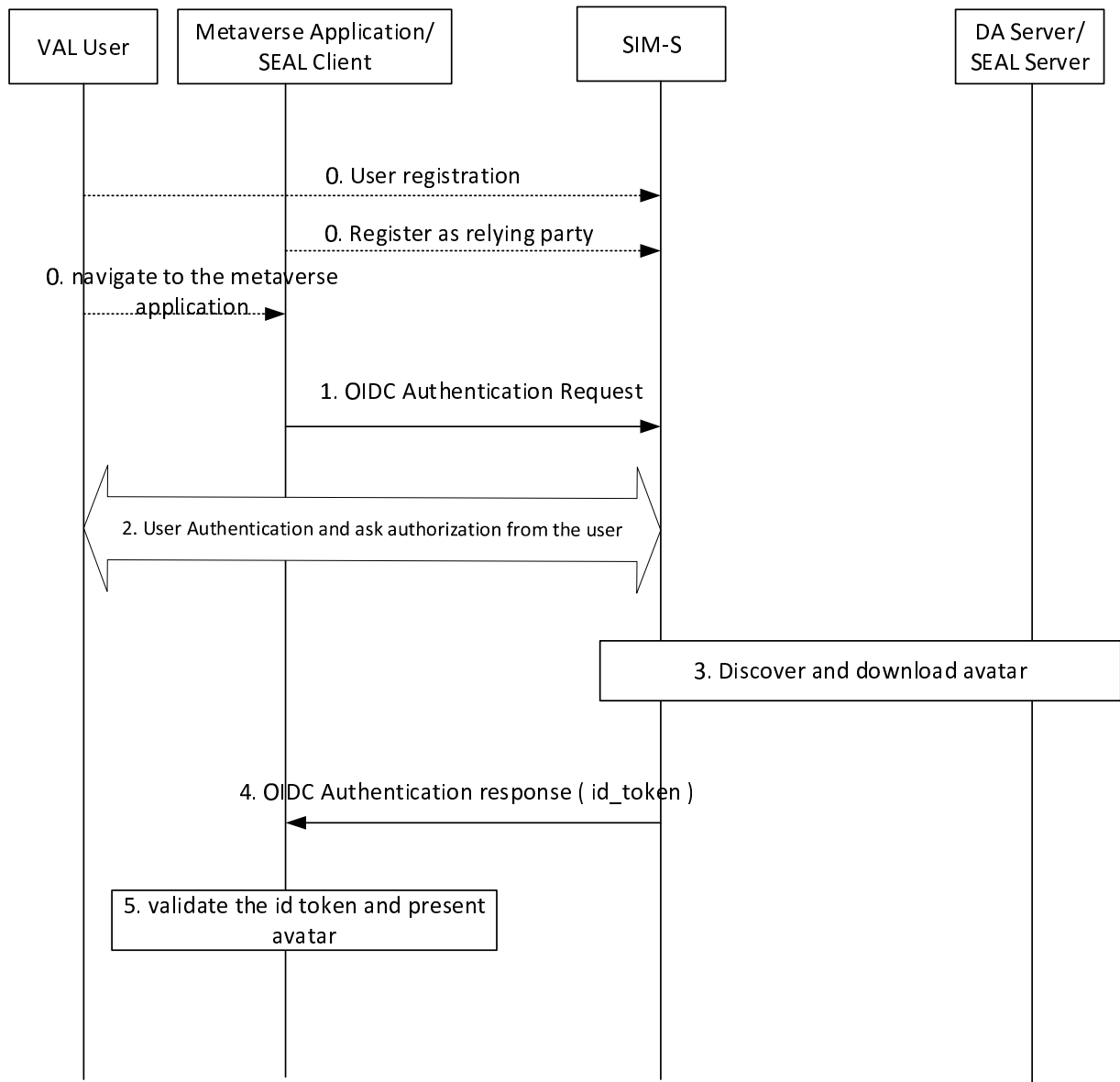


Figure 6.3.2.3-1. Procedure to authorize a VAL user to use the avatar in metaverse service

0. DA client registers to SIM-S as Relying Party and delegates identity management and user authentication to the SIM-S, which takes the role of OpenID Provider. The VAL User registers to the SIM-S with user profile including name and gets User ID from the SIM-S. The VAL User navigates to a metaverse application with the User ID.

NOTE 1: The step 0 is implementation dependent and out of 3GPP scope.

NOTE 2: As central identity management server, SIM-S is trusted by VAL User, metaverse application and DA server, and the SIM-S is authorized to retrieve avatar of the VAL user from the DA server. Security for the SEAL-E specified in TS 33.434 [4] is applied for protection of interface between SIM-S and DA server, which including mutual authentication, confidentiality and integrity protection.

1. With the User ID, the metaverse application sends OIDC authentication request to the SIM-S, including User ID and response type. The Scope is set to open id, name and avatar, the response type is set to id_token.
2. The SIM-S triggers user authentication towards the VAL User and gets authorization from the VAL User to allow the SIM-S to download his/her avatar and send to the metaverse application.

3. The SIM-S discovers an avatar for the VAL User in the metaverse application and downloads the avatar from the DA server.
4. The SIM-S sends authentication response to the metaverse application. The response includes an ID token which comprises of issuer set to the SIM-S, audience set to the metaverse application, subject set to the User ID, and name claim set to the name of the VAL User, and additional claims set to avatar media of the VAL User, and optional avatar ID and avatar name.
5. The metaverse application validates the ID token to authenticate the VAL User, and retrieves the name and avatar of the user from the ID token and displays the information.

6.4 Privacy protection for user information exposure

When CAPIF is used as specified in TS 23.434 [2], during exposure of user specific information (e.g. user identity, user location) in localized mobile metaverse services through the application enabler layer, the RNAA framework defined in TS 33.122 [29] is used for privacy protection of user information. The user information is only exposed if CCF obtains permission from the resource owner as specified in TS 33.122 [29].

When CAPIF is not used, the user information is only exposed if SIM-S obtains authorization decision from the VAL user after the SIM-S authenticates the VAL user based on OpenID connection (OIDC) procedure referred in TS 33.434 [4].

NOTE 1: The above procedures are only applicable to the case where the resource owner is same to the VAL user.

NOTE 2: How CAPIF RNAA supports service operation level and resource level authorization is specified in TS 33.122 [29].

Annex A (normative): OpenID connect profile for VAL

A.1 General

The information in this annex provides a normative description of the Authentication and Authorization framework based on the OpenID Connect 1.0 standard. Characterization of the ID token, access token, how to obtain tokens, how to validate tokens, and how to use the refresh token is explained.

The OpenID Connect 1.0 standard provides the source of the information contained in this annex. This annex profiles the OpenID Connect standard and includes the ID token and the access token, as well as the definition of VAL specific scopes for key management, VAL services, configuration management, and group management. This profile is compliant with OpenID Connect.

A.2 VAL tokens

A.2.1 ID token

A.2.1.1 General

The ID Token shall be a JSON Web Token (JWT) and contain the following standard and VAL token claims. Token claims provide information pertaining to the authentication of the VAL client by the SIM-S as well as additional claims. The following clause profiles the required standard and VAL claims for the VAL Connect profile.

A.2.1.2 Standard claims

These standard claims are defined by the OpenID Connect 1.0 specification and are REQUIRED for VAL implementation. Other claims defined by OpenID Connect are optional. The standards-based claims for a VAL Connect ID token are shown in table A.2.1.2-1.

Table A.2.1.2-1: ID token standard claims

Parameter	Description
Iss	REQUIRED. The URL of the SIM-S.
Sub	REQUIRED. A case-sensitive, never reassigned string (not to exceed 255 bytes), which uniquely identifies the VAL user within the VAL server provider's domain.
Aud	REQUIRED. The Oauth 2.0 client_id of the SIM-C
Exp	REQUIRED. Implementers MAY provide for some small leeway, usually no more than a few minutes, to account for clock skew (not to exceed 30 seconds)
iat	REQUIRED. Time at which the ID Token was issued. Its value is a JSON number representing the number of seconds from 1970-01-01T0:0:0Z as measured in UTC until the date/time.

A.2.1.3 VAL claims

The VAL Connect profile extends the OpenID Connect standard claims with the additional claims based on the VAL service.

A.2.2 Access token

A.2.2.1 Introduction

The access token is opaque to VAL clients and is consumed by the VAL resource servers. The access token shall be encoded as a JSON Web Token as defined in IETF RFC 7797 [11]. The access token shall include the JSON web digital signature profile as defined in IETF RFC 7515 [12].

A.2.2.2 Standard claims

VAL access tokens shall convey the following standards-based claims as defined in IETF RFC 7662 [13].

Table A.2.2.2-1: Access token standard claims

Parameter	Description
Exp	REQUIRED. Implementers MAY provide for some small leeway, usually no more than a few minutes, to account for clock skew (not to exceed 30 seconds).
Scope	REQUIRED. A JSON string containing a space-separated list of the authorization scopes associated with this token. The scope(s) contained here reflect the requested scope(s) from the Authentication Request (clause A.4.2.2).
client_id	REQUIRED. The identifier of the SIM-C making the API request as previously registered with the SIM-S.

A.2.2.3 VAL claims

The VAL profile extends the standard claims defined in IETF RFC 7662 [13] with the additional claims based on the VAL service and those shown in table A.2.2.3-1.

Table A.2.2.3-1: Access token VAL claims

Parameter	Description
SKeyProv	OPTIONAL for SEAL. The SKeyProv parameter shall be present when the VAL Server SKM-C is authorized to provide key material to the KMS.

A.3 SIM-C registration

Before a SIM-C can obtain ID tokens and access tokens (required to access VAL resource servers) it shall first be registered with the SIM-S of the service provider as required by OpenID Connect 1.0. The method by which this is done is not specified by this profile. For native SIM-C, the following information shall be registered:

- The client is issued a client identifier. The client identifier represents the client's registration with the authorization server, and enables the SIM-S to reference parameters associated with that client's registration when being requested for an access token by the SIM-C.
- Registration of the client's redirect URIs.

Other information about the SIM-C such as (for example): application name, website, description, logo image, legal terms to be consented to, may optionally be registered.

A.4 Obtaining tokens

A.4.1 General

Once a SIM-C has been successfully registered with the SIM-S of the VAL service provider, the SIM-C may request ID tokens and access tokens (as required to access VAL service servers). Only native SIM-C are defined here. The exact

method in which a SIM-C requests the access token depends upon the client profile. The SIM-C profiles, along with steps required from them to obtain OAuth access tokens, are explained below.

A.4.2 Native SIM-C

A.4.2.1 General

This conforms to the Native Application profile of OAuth 2.0 as per IETF RFC 6749 [3].

SIM-C fitting the Native application profile utilize the authorization code grant type with the PKCE extension for enhanced security as shown in figure A.4.2.1-1.

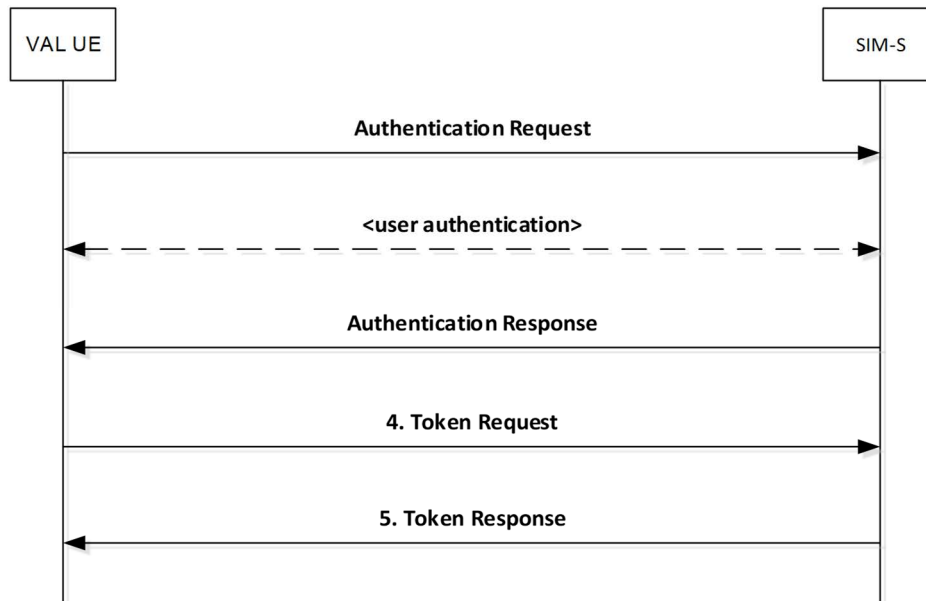


Figure A.4.2.1-1: Authorization Code flow

A.4.2.2 Authentication request

As described in OpenID Connect 1.0, the SIM-C constructs a request URI by adding the following parameters to the query component of the authorization endpoint's URI using the "application/x-www-form-urlencoded" format, redirecting the user's web browser to the authorization endpoint of the SIM-S. The standard parameters shown in table A.4.2.2-1 are required by this Connect profile. Other parameters defined by the OpenID Connect specification are optional.

Table A.4.2.2-1: Authentication Request standard required parameters

Parameter	Values
response_type	REQUIRED. For native SIM-C the value shall be set to "code".
client_id	REQUIRED. The identifier of the SIM-C making the API request. It shall match the value that was previously registered with the SIM-S of the VAL service provider.
Scope	REQUIRED. Scope values are expressed as a list of space-delimited, case-sensitive strings which indicate which VAL resource servers the client is requesting access to. If authorized, the requested scope values will be bound to the access token returned to the client. The scope value "openid" is defined by the OpenID Connect standard and is mandatory, to indicate that the request is an OpenID Connect request, and that an ID token should be returned to the SIM-C. NOTE: Additional VAL service specific scopes need to be defined by VAL service specification and it is out of scope of the present document.
redirect_uri	REQUIRED. The URI of the SIM-C to which the SIM-S will redirect the SIM-C's user agent in order to return the authorization code to the SIM-C. The URI shall match the redirect URI registered with the SIM-S during the client registration phase.
State	REQUIRED. An opaque value used by the SIM-C to maintain state between the authorization request and authorization response. The SIM-S includes this value in its authorization response back to the SIM-C.
acr_values	REQUIRED. Space-separated string that specifies the acr values that the SIM-S is being requested to use for processing this authorization request, with the values appearing in order of preference. For minimum interoperability requirements, a password-based ACR value is mandatory to support. "3gpp:acr:password".
code_challenge	REQUIRED. The base64url-encoded SHA-256 challenge derived from the code verifier that is sent in the authorization request, to be verified against later.
code_challenge_method	REQUIRED. The hash method used to transform the code verifier to produce the code challenge. This profile current requires the usage of "S256"
NOTE: The order in which they are expressed does not matter.	

A.4.2.3 Authentication response

The authorization endpoint running on the SIM-S issues an authorization code and delivers it to the SIM-C. The authorization code is used by the SIM-C to obtain an ID token, access token and refresh token from the SIM-S. The authorization code is added to the query component of the redirection URI using the "application/x-www-form-urlencoded" format. The authorization code standard parameters are shown in table A.4.2.3-1.

Table A.4.2.3-1: Authentication Response standard required parameters

Parameter	Values
Code	REQUIRED. The authorization code generated by the authorization endpoint and returned to the SIM-C via the authorization response.
State	REQUIRED. The value shall match the exact value used in the authorization request. If the state does not match exactly, then the NGMI API client is under a Cross-site request forgery attack and shall reject the authorization code by ignoring it and shall not attempt to exchange it for an access token. No error is returned.

A.4.2.4 Access token request

In order to exchange the authorization code for an ID token, access token and refresh token, the SIM-C makes a request to the authorization server's token endpoint by sending the following parameters using the "application/x-www-form-urlencoded" format, with a character encoding of UTF-8 in the HTTP request entity-body. Note that client authentication is REQUIRED for native applications (using PKCE) in order to exchange the authorization code for an access token. Assuming that client secrets are used, the client secret is sent in the HTTP Authorization Header. The access token request standard parameters are shown in table A.4.2.4-1.

Table A.4.2.4-1: Access token request standard required parameters

Parameter	Values
grant_type	REQUIRED. The value shall be set to "authorization_code".
code	REQUIRED. The authorization code previously received from the SIM-S as a result of the authorization request and subsequent successful authentication of the VAL user.
client_id	REQUIRED. The identifier of the client making the API request. It shall match the value that was previously registered with the OAuth Provider during the client registration phase of deployment, or as provisioned via a development portal.
redirect_uri	REQUIRED. The value shall be identical to the "redirect_uri" parameter included in the authorization request.
code_verifier	REQUIRED. A cryptographically random string that is used to correlate the authorization request to the token request.

A.4.2.5 Access token response

If the access token request is valid and authorized, the SIM-S returns an ID token, access token and refresh token to the SIM-C in an access token response message; otherwise it will return an error.

The access token response standard parameters are shown in table A.4.2.5-1.

Table A.4.2.5-1: Access token response standard parameters

Parameter	Values
access_token	REQUIRED. This is the issued access token.
token_type	REQUIRED. This field shall be "bearer"
expires_in	REQUIRED. The lifetime in seconds of the access token.
Id_token	OPTIONAL. This is the issued id token.
Refresh_token	OPTIONAL. This is the issued refresh token.

The SIM-C may now validate the user with the ID token and configure itself for the user (e.g. by extracting the VAL service ID from the ID Token). The SIM-C then uses the access token to make authorized requests to the SIM resource servers on behalf of the end user.

A.5 Refreshing an access token

A.5.1 General

To protect against leakage or other compromise, access token lifetimes are typically short lived (though it is ultimately a matter of security policy & configuration by the service provider). Some client types can be issued longer-lived refresh tokens, which enable them to refresh the access token and avoid having to prompt the user for authentication again when the access token expires. Refresh tokens are available only to clients utilizing the authorization code grant type. Figure A.5.1-1 shows how Native SIM-C can use the refresh token as a grant type to obtain new access tokens.



Figure A.5.1-1: Requesting a new access token

A.5.2 Access token request

To obtain an access token from the SIM-S using a refresh token, the SIM-C makes an access token request to the token endpoint of the SIM-S. The SIM-C does this by adding the following parameters using the "application/x-www-form-urlencoded" format, with a character encoding of UTF-8 in the HTTP request entity-body. The access token request standard parameters are shown in table A.5.2-1.

Table A.5.2-1: Access token request standard required parameters

Parameter	Values
grant_type	REQUIRED. The value shall be set to "refresh_token".
Scope	Space-delimited set of permissions that the SIM-C requests. Note that the scopes requested using this grant type shall be of equal to or lesser than scope of the original scopes requested by the SIM-C as part of the original authorization request.

If the SIM-C was provided with client credentials by the SIM-S, then the client shall authenticate with the token endpoint of the SIM-S utilizing the client credential (shared secret or public-private key pair) established during the client registration phase.

A.5.3 Access token response

In response to the access token request (above) the token endpoint on the SIM-S will return an access token to the SIM-C, and optionally another refresh token in an access token response message.

The access token response standard parameters are shown in table A.5.3-1.

Table A.5.3-1: Access token response standard parameters

Parameter	Values
access_token	REQUIRED. This is the issued access token.
token_type	REQUIRED. This field shall be "bearer"
expires_in	REQUIRED. The lifetime in seconds of the access token.
Id_token	OPTIONAL. This is the issued id token.
Refresh_token	OPTIONAL. This is the issued refresh token.

It is possible to configure the SIM-S to confirm that the user account is still valid each time the refresh token is presented, and to revoke the refresh token if not. This security practice is RECOMMENDED.

A.6 Using the token to access VAL resource servers

Connect for VAL shall initially support the bearer access token type. Access tokens of type "bearer" shall be communicated from the VAL or SEAL Clients in UE to VAL resource servers by including the access token in the HTTP Authorization Header, per IETF RFC 6750 [4].

The access token is opaque to the VAL or SEAL Clients in UE, meaning that the client does not have any knowledge of the access token itself. The client will be given some metadata corresponding to the access token, such as its expiration time, so that it does not send an expired access token to VAL resource servers. If the access token is presented to a VAL resource server and the scope is invalid or the token is expired or revoked, the VAL resource server should return an error message indicating such to the VAL or SEAL Clients in UE.

A.7 Token validation

A.7.1 ID token validation

The VAL or SEAL Clients in UE shall validate the ID token as per clause 3.1.3.7 of the OpenID Connect 1.0 specification [5].

A.7.2 Access token validation

VAL resource servers shall validate access tokens received from the VAL or SEAL Clients in UE according to IETF RFC 7797 [11].

A.8 Token revocation

In order to limit the time validity of a token, the "exp" and "expires_in" parameters may be used as a method of access token revocation. If either the "exp" or "expires_in" parameter is used as a method of access token revocation, then the following applies:

Within the standard claims of an access token, the "exp" parameter shall be used by the authorising server to determine whether or not the token is valid. If the current time is beyond the time specified by the "exp" parameter, the associated token shall no longer be considered valid and any requests made with an expired token shall be rejected by the authorising server.

Within the standard claims of an access token response, token exchange response or token response message, the "expires_in" parameter shall be used by the UE client(s) to determine validity of the associated token. If the current time is beyond the time specified by the "expires_in" parameter, the associated token shall no longer be considered valid and no client requests shall be made using the expired token. A refresh token may be used per clause A.5 to obtain a new access token.

A.9 SIM-S interface security

The support of Transport Layer Security (TLS) between the SIM-C in the VAL UE and the SIM-S is mandatory. The profile for TLS implementation and usage shall follow the provisions given in 3GPP TS 33.310 [6], annex E.

If PSK TLS based authentication is supported, the SIM-C in the VAL UE and the SIM-S shall support the TLS version, PSK ciphersuites and TLS Extensions as specified in the TLS profile given in 3GPP TS 33.310 [6], annex E. The usage of pre-shared key ciphersuites for TLS is specified in the TLS profile given in 3GPP TS 33.310 [6], annex E.

Annex B (normative): Security mechanisms for LWP interfaces

B.1 General

This annex specifies communication security, authentication and authorization mechanisms for protocol realizations of the light-weight protocol (LWP) in the signalling control plane.

B.2 Communication security for CoAP

CoAP messages [18] shall be protected and deploy the security enhancements of [22]. When (D)TLS is used, the (D)TLS and certificate profiling shall follow TS 33.210 [14] and TS 33.310 [6]. When OSCORE is used, the mandatory to implement provisions given by RFC 8613 [23] shall be followed.

B.3 Authentication and authorization mechanism on CoAP

B.3.1 General

When CoAP is used for the LWP, Authentication and authorization for Constrained Environments (ACE) using OAuth 2.0 Framework (ACE-OAuth) as specified in [19] shall be supported.

Figure B.3.1-1 shows the functional model which consists of the SEAL Identity Management Server (SIM-S), SEAL Identity Management Client (SIM-C) and SEAL server. The IM-UU reference point between the SIM-S and the SIM-C and the SEAL-UU reference point between SEAL server and SIM-C shall support ACE-OAuth [19] and OAuth 2.0 [9] with COSE [20].

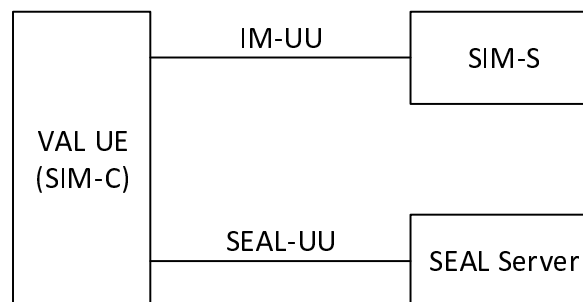


Figure B.3.1-1: Functional model for SEAL Identity management client, server and SEAL server

The SIM-S, the SIM-C and a SEAL server respectively play the roles of the Authorization Server, the Client and the Resource Server in the ACE-OAuth framework.

For authentication of SIM-S, the security enhancements of CoAP specified in [22] shall be followed. When (D)TLS is used, the (D)TLS and certificate profiling shall follow TS 33.210 [14] and TS 33.310 [6]. When OSCORE is used, authentication shall be based on pre-shared secrets. The authentication method and credentials of the VAL-UE are out of scope of this specification.

B.3.2 VAL user authentication

VAL user authentication is executed by the SIM-S before providing access token for the VAL UE.

NOTE 1: The primary method and credentials for VAL user authentication are based on VAL service provider policy. The method chosen by the VAL service provider for authentication and authorization is neither defined nor limited by the present document.

NOTE 2: VAL user may refer to the user or the device depending on the setting.

B.3.3 SEAL service authorization

SEAL Service Authorization procedure shall validate the VAL user to access the SEAL services. In order to gain access to SEAL services, the SEAL client shall present an access token to the SEAL server for each service of interest. If the access token is valid, then the client shall be granted to use the service.

B.3.4 Authorization framework

Authorization framework is shown in figure B.3.4-1. The ACE-OAuth [19] framework is followed. The SIM-S and SIM-C shall perform mutual authentication as specified in B.3.1. After successful authentication, the SIM-C shall request and receive an access token from the SIM-S over CoAP as described in clause 5.8 of [19] indicated in steps 1 and 2 in the figure. Before providing the access token, SIM-S shall authorize the VAL UE for the requested service. The procedures may be repeated as necessary to obtain additional VAL UE authorizations.

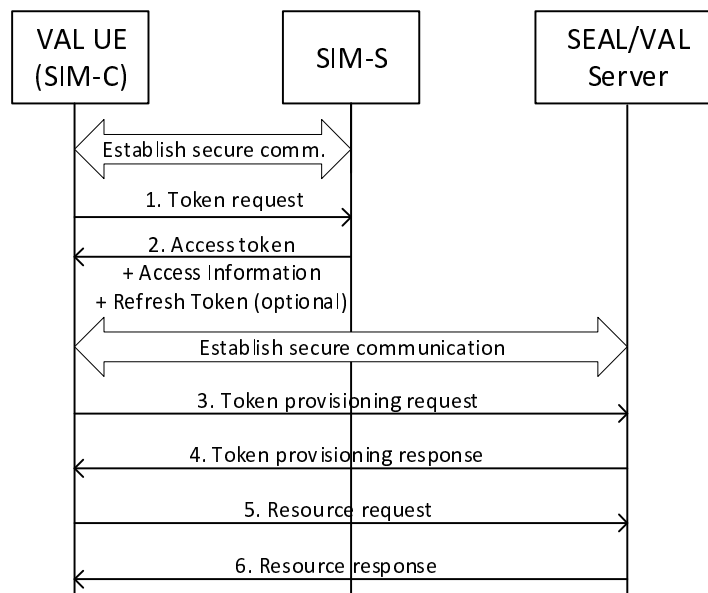


Figure B.3.4-1: VAL UE Service Authorization

After the VAL UE received an access token it shall establish a secure connection with the SEAL/VAL server as specified in B.2. The VAL UE shall send a CoAP message containing the access token to the SEAL/VAL server in a service authorization request as described in clause 5.10 of [19] indicated in steps 3 and 4 in the figure. On receiving the service authorization message, the SEAL/VAL server shall validate the access token. If the access token is valid, the SEAL/VAL server shall provide service-related information according to the rights granted to the VAL UE in response to subsequent requests indicated in steps 5 and 6.

The messages sent for the authorization shall be protected. When (D)TLS is used, the (D)TLS and certificate profiling shall in addition to [22] follow also TS 33.210 [14] and TS 33.310 [6]. When the VAL UE is authenticating directly to the SEAL/VAL server, then the DTLS or TLS profile of ACE [21,25] may be used. In order to authorize clients and protect communication across proxies, the OSCORE profile of ACE [24] shall be used.

B.3.5 VAL service authorization

The VAL service authorization procedure shall validate the VAL user authorized to access the VAL services. In order to gain access to VAL services, the VAL client shall present an access token to the VAL server for each VAL service of interest (see clause B.3.4). If the access token is valid, then the VAL client shall be granted use of the requested VAL service.

B.3.6 Access token

B.3.6.1 Introduction

The access token is opaque to VAL clients and is consumed by the VAL resource servers. The access token shall be encoded as a CBOR Web Token as defined in IETF RFC 8392 [26]. Depending on whether the CWT is signed, MACed or encrypted, the corresponding COSE object shall be used as defined in IETF RFC 8392 [26].

B.3.6.2 Standard claims

VAL access tokens shall convey the following standards-based claims as specified in ACE-OAuth [19].

Table B.3.6.2-1: Access token standard claims

Parameter	Description
exp	REQUIRED. Implementers MAY provide for some small leeway, usually no more than a few minutes, to account for clock skew (not to exceed 30 seconds).
scope	REQUIRED. Text or byte string. The text string contains a space-separated list of the authorization scopes associated with this token. The byte string allows compact encoding of complex scopes. The scope(s) contained here reflect the requested scope(s) from the Token Request (clause B.3.4).
cnf	REQUIRED. The "cnf" (confirmation) claim declares that the SEAL client possesses a particular key and that the SEAL service can cryptographically confirm that the SEAL client has possession of that key. The value of the "cnf" claim is a CBOR map and the members of that map identify the proof-of-possession key [27].
audience	OPTIONAL. This field indicates the targeted SEAL servers/resources for the access token [19].

B.3.6.3 VAL claims

The VAL profile extends the standard claims specified in ACE-OAuth [19] with the additional claims based on the VAL service.

B.3.7 Obtaining access tokens

B.3.7.1 Access token request

In order to obtain an access token (and optionally a refresh token) the SEAL client makes a CoAP request to the authorization server's token endpoint by sending the following parameters using the "application/ace+cbor" Content-format, with a CBOR map in the CoAP payload. Note that mutual authentication is REQUIRED between SEAL client and SEAL server. The access token request standard parameters are shown in table B.3.7.1-1.

Table B.3.7.1-1: Access token request standard required parameters

Parameter	Values
scope	OPTIONAL. This field requests authorization scopes for the access token.
audience	OPTIONAL. This field requests specific SEAL servers/resources for the access token [19].
cnonce	REQUIRED and only used if a client-nonce was provided in response to an unauthorized resource request to a SEAL server/resource [19].
req_cnf	OPTIONAL. This field contains information about the key the SEAL client wants to bind to the access token for proof-of-possession [28].

B.3.7.2 Access token response

If the access token request is valid and authorized, the SEAL server returns an access token (and optionally a refresh token) to the SEAL client in an access token response message; otherwise, it will return an error.

The access token response standard parameters are shown in table B.3.7.2-1.

Table B.3.7.2-1: Access token response standard parameters

Parameter	Values
access_token	REQUIRED. This is the issued access token.
expires_in	REQUIRED. The lifetime in seconds of the access token.
refresh_token	OPTIONAL. This is the issued refresh token.
ace_profile	REQUIRED. This field indicates the IETF ACE profile the SEAL client shall use towards the SEAL server/resource [19].
cnf	OPTIONAL. This field is REQUIRED for symmetric key usages unless the secret key is known to the SEAL client (e.g. in case of update of access rights) [27].
rs_cnf	OPTIONAL. This field is REQUIRED for asymmetric key usages unless the public key of the SEAL server is known to the SEAL client (e.g. in case of update of access rights) [28].

The SEAL client may now use the access token to make protected and authorized requests to the SEAL server.

Annex C (informative): Change history

Change history							
Date	Meeting	TDoc	CR	Rev	Cat	Subject/Comment	New version
2020-07	SA#88e					Upgrade to change control version	16.0.0
2020-09	SA#89e	SP-200715	0001	-	F	KM Clarifications	16.1.0
2020-09	SA#89e	SP-200715	0002	1	F	TS 33.434 clean up	16.1.0
2021-03	SA#91e	SP-210112	0003	1	F	correction in clause 5.2.4	16.2.0
2021-12	SA#94e	SP-211367	0004	1	B	Security for CoAP interfaces in SEAL	17.0.0
2022-03	SA#95e	SP-220218	0005	-	B	Updating SEAL-S security	17.1.0
2022-03	SA#95e	SP-220218	0006	-	B	Updating SEAL-UU security	17.1.0
2022-03	SA#95e	SP-220218	0007	-	B	Profiling ACE in SEAL	17.1.0
2022-03	SA#95e	SP-220218	0008	1	F	Correcting the implementation of approved S3-214431 to SEAL TS 33.434	17.1.0
2022-03	SA#95e	SP-220224	0010	1	A	Rel-17 CAPIF usage for SEAL-S	17.1.0
2022-03	SA#95e	SP-220224	0012	1	A	Rel-17 Correcting SEAL-UU security	17.1.0
2022-03	SA#95e					Adding a paragraph missing from implementation of 0012r1	17.1.1
2022-06	SA#96	SP-220459	0013	-	F	Updates to 33.434 for CoAP usage	17.2.0
2022-09	SA#97e	SP-220880	0014	-	F	Correction on the implementation of CR 0013 (S3-220917)	17.3.0
2023-06	SA#100	SP-230612	0015	1	B	Add security aspect of SEAL Data Delivery enabler	18.0.0
2023-06	SA#100	SP-230613	0016	-	B	SEAL security for network domain interfaces	18.0.0
2023-09	SA#101	SP-230906	0017	1	B	Key Provisioning procedure	18.1.0
2024-03	SA#103	SP-240354	0019	1	A	Updating Internet Drafts to Final RFCs (Rel-18)	18.2.0
2025-06	SA#108	SP-250659	0021	-	B	AIMLE Security Alignment	19.0.0
2025-06	SA#108	SP-250681	0023	-	B	Security procedures for mobile metaverse services	19.0.0

History

Document history		
V19.0.0	January 2026	Publication