ETSI TS 124 542 V18.0.1 (2024-06)

5G;
Notification management - Service Enabler Architecture Layer for Verticals (SEAL);
Protocol specification
(3GPP TS 24.542 version 18.0.1 Release 18)

Reference
DTS/TSGC-0124542vi01

Keywords
5G

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

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 Web server (https://ipr.etsi.org/).

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™** and **LTE™** 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 under https://webapp.etsi.org/key/queryform.asp.

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

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 protocol aspects for the notification management capability of SEAL to support vertical applications (e.g. V2X) over the 3GPP system for NM-UU reference point.

The present document is applicable to the User Equipment (UE) supporting the notification management client functionality as described in 3GPP TS 23.434 [2], to the application server supporting the notification management server functionality as described in 3GPP TS 23.434 [2] and to the application server supporting the vertical application server (VAL server) functionality as defined in specific vertical application service (VAL service) specification.

NOTE: The specification of the VAL server for a specific VAL service is out of scope of the present document.

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] 3GPP TS 24.547: "Identity management - Service Enabler Architecture Layer for Verticals (SEAL); Protocol specification".

[4] IETF RFC 6750: "The OAuth 2.0 Authorization Framework: Bearer Token Usage".

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

3 Definitions of terms, symbols and abbreviations

3.1 Terms

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

SEAL notification management client: An entity that provides the client side functionalities corresponding to the SEAL notification management service.

SEAL notification management server: An entity that provides the server side functionalities corresponding to the SEAL notification management service.

For the purposes of the present document, the following terms and definitions given in 3GPP TS 23.434 [2] apply:

SEAL client
SEAL server
SEAL service
VAL notification
VAL server
VAL service

VAL user
Vertical
Vertical application

3.2 Abbreviations

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

SEAL	Service Enabler Architecture Layer for verticals
SNM-C	SEAL Notification Management Client
SNM-S	SEAL Notification Management Server
VAL	Vertical Application Layer

4 General description

Notification management is a SEAL service that provides the notification management related capabilities to one or more vertical applications. The present document enables a SEAL notification management Client (SNM-C) and VAL server to manage notifications through the SEAL notification management Server (SNM-S).

5 Functional Entities

5.1 SEAL notification management client (SNM-C)

The SNM-C is a functional entity that acts as the application client for notification management.

To be compliant with the HTTP procedures in the present document, the SNM-C:

a) shall support the procedure of creating and opening notification channel as per clause 6.2.2.1;

b) shall support the procedure to receive PUSH notification message from the SNM-S as per clause 6.2.3.1;

c) shall support the procedure to PULL notification message from the SNM-S as per clause 6.2.3.X.1;

d) shall support the procedure to update notification channel as per clause 6.2.5.1; and

e) shall support the procedure to delete notification channel as per clause 6.2.4.1.

5.2 SEAL notification management server (SNM-S)

The SNM-S functional entity provides notification management support within the vertical application layer.

To be compliant with the HTTP procedures in the present document, the SNM-S:

a) shall support the procedure of creating and opening notification channel as per clause 6.2.2.2;

b) shall support the procedure to send PUSH notification message to the SNM-C as per clause 6.2.3.2;

c) shall support the procedure to PULL notification message as per clause 6.2.3.X.2;

d) shall support the procedure to update notification channel as per clause 6.2.5.2; and

e) shall support the procedure to delete notification channel as per clause 6.2.4.2.

6 Notification management procedures

6.1 General

6.2. On-network procedures

6.2.1 General

6.2.1.1 Authenticated identity in HTTP request

Upon receiving an HTTP request, the SNM-S shall authenticate the identity of the sender of the HTTP request as specified in 3GPP TS 24.547 [3], and if authentication is successful, the SNM-S shall use the identity of the sender of the HTTP request as an authenticated identity.

6.2.1.2 Boot up procedure

Upon device boot up, the NM-C in the UE shall create the notification channel with the notification management server as specified in clause 6.2.2.1.

6.2.2 Notification channel creation procedure

6.2.2.1 SNM client procedures

Upon receiving a request from VAL service to receive notifications via the notification channel; the SNM-C may create a notification channel by sending an HTTP POST request to the SNM-S. In the HTTP POST request the SNM-C:

a) shall set the Request-URI to the URI of the SNM-S;

b) shall include the Host header with public user identity of SNM-S;

c) shall include an Authorization header field with the "Bearer" authentication scheme set to an access token of the "bearer" token type as specified in IETF RFC 6750 [4];

d) shall include a Content-Type header field set to "application/vnd.3gpp.seal-create-notification-channel-request";

e) shall generate the create notification channel request message as specified in clause A.1.2:

   1) shall set the requestor identity to the notification management client identity;

   2) shall set the channel type to PULL or PUSH based on the VAL Application requesting the use of notification channel;

   3) may set the PUSH channel details parameter with the PUSH callback-URL if the channel type is PUSH;

   4) shall set the validity duration of the notification channel;

   5) may set the pull notification message trigger parameter to "true", if in case the SNM-C support application trigger to initiate pull notification message procedure; and

NOTE: Application trigger enables SNM-C to pull notification messages from SNM-S only when the outstanding notifications are available at SNM-S.

   6) shall set the VAL id cluster list parameter with list of VAL identities corresponding to each VAL service requesting to receive notifications via the notification channel; and

f) include the parameters specified in clause A.1.2 serialized into a JavaScript Object Notation (JSON) structure as specified in IETF RFC 7159 [5].

Upon receiving an HTTP 200 (OK), the SNM-C shall notify the VAL services about the successful creation of notification channel and shall listen on the PUSH callback-URL for receiving the PUSH notification messages from SNM-S.

6.2.2.2 SNM server procedures

Upon reception of an HTTP POST request from SNM-C where the Request-URI of the HTTP POST request contains the URI of the SNM-S, the SNM-S:

a) shall determine the requestor identity of the received HTTP POST request as specified in clause 6.2.1.1, and:

   1) if the identity of the sender of the received HTTP POST request is not authorized user, shall respond with an HTTP 403 (Forbidden) response to the HTTP POST request and skip rest of the steps;

b) shall process the create notification channel request and if the channel type is:

   1) PUSH, the SNM-S shall process:

      i) the PUSH channel details parameter as specified in clause A.1.2 to fetch the PUSH callback-URL of the SNM-C:

         A) if the PUSH callback-URL is not provided the SNM-S shall respond with an HTTP 406 (Not Acceptable) response to the HTTP POST request and skip rest of the steps; and

         B) the SNM-S shall store this PUSH callback-URL for future usages, when the notification messages are to be pushed to SNM-C; and

      ii) the VAL id cluster list parameter, which is a list of VAL identities corresponding to each VAL services requesting to receive notifications messages via the notification channel; and

   2) PULL, the SNM-S:

      i) may process the pull notification message trigger attribute if provided. If the SNM-S supports sending the application trigger to SNM-C to initiate pulling notification message procedure, then the SNM-S may share this indication in the create notification channel response and store this for future usage to send triggers to SNM-C for outstanding notification messages received from VAL server; and

NOTE 1: The SNM-S can utilize the application triggering services (or Device Triggering) provided by the 3GPP core network via NEF or SCEF.

      ii) shall wait for the SNM-C to pull the notification messages; and

Editor's note: In case of multiple notification accumulated at the SNM-S from same VAL Server towards the SNM-C. How the SNM-S decides to share all the notifications or latest notifications is FFS.

c) shall process the validity duration share by the SNM-C.

NOTE 2: The SNM-S can store the authorized user and information shared as part of create notification channel request for future references

Upon successful creation of notification channel; the SNM-S:

a) shall create a notification channel response message with below attributes as specified in clause A.1.3;

   1) shall generate unique channel identifier;

   2) shall generate the callback URL, which shall be used by VAL clients in UE for sharing it to VAL Server as part of their respective services;

   3) may generate the validity duration of the notification channel; and

   4) may generate a notification URL that shall be used by SNM-C to pull the notifications from SNM-S in case of PULL channel type;

b) shall include a Content-Type header field set to "application/vnd.3gpp.seal-create-notification-channel-response"; and

c) shall send an HTTP 200 (OK) response including message generated above.

6.2.3 Notification Message delivery

6.2.3.1 PUSH notification messages procedure

6.2.3.1.1 SNM client procedure

Upon receiving the HTTP POST request over a call back URI which was given to SNM-S at the time of notification channel creation, the SNM-C:

a) shall match identifier received in the channel identifier parameter of the HTTP POST request with the locally stored channel identifier. If channel identifier is not matching, then:

   1) send an HTTP 406 (Not Acceptable) response to SNM-S and skip rest of the steps;

b) shall send an HTTP 200 (OK) response to SNM-S; and

c) shall process the VAL notification message list parameter received in HTTP request entity-body as specified in clause A.2.2 and deliver each received notification message to the appropriate VAL client on UE that matches the VAL id cluster info parameter received with each message.

6.2.3.1.2 SNM server procedure

To send the PUSH notification messages received from the VAL server to the SNM-C, the SNM-S:

a) shall check whether an PUSH notification channel exists with the SNM-C to receive the notification messages matching to the VAL identities (VAL UE or VAL user ID, VAL service ID, VAL application ID) shared by VAL server; if there is no channel created then skip rest of the steps;

b) shall generate an HTTP POST message to send notification messages received from the VAL server. In the HTTP POST message:

   1) shall set request URI to the call back URI received at the time of creating channel;

   2) shall set Content-Type header to "application/vnd.3gpp.seal-notification-payload/json";

   3) shall generate the notification message payload as specified in clause A.2.2:

      i) shall set the channel identifier associated with the SNM-C; and

      ii) shall generate the notification message list for the messaged received from VAL servers as specified in clause A.2.2-2; and

   4) shall include an HTTP request entity-body with the parameters specified in clause A.2.2 serialized into a JavaScript Object Notation (JSON) structure as specified in IETF RFC 7159 [5]; and

c) shall send the HTTP POST request towards SNM-C.

6.2.3.2 PULL notification messages procedure

6.2.3.2.1 SNM client procedure

To retrieve the latest notification messages available at the SNM-S as received from VAL servers, the SNM-C shall send an HTTP GET request to the SNM-S. In the HTTP GET request the SNM-C:

a) shall set the Request-URI with the "notification URL" received in create notification channel response message as specified in clause A.1.3;

b) shall include the Host header with public user identity of SNM-S;

c) shall include an Authorization header field with the "Bearer" authentication scheme set to an access token of the "bearer" token type as specified in IETF RFC 6750 [4];

d) shall include a Content-Type header field set to "application/vnd.3gpp.seal-pull-notification-message-request/json";

e) shall generate the pull notifications request message as specified in clause A.2.3:

1) shall set the requestor identity to the notification management client identity; and

2) shall set the channel identifier to the identity of the corresponding notification channel with SNM-S from which the messages has to be pulled.

3) include the parameters specified in clause A.2.3 serialized into a JavaScript Object Notation (JSON) structure as specified in IETF RFC 7159 [5].

Upon receiving an HTTP 200 (OK), the SNM-C shall parse the notification messages received from SNM-S as specified in table A.2.2-1 and notify the VAL services. The SNM-C shall immediately send HTTP GET towards the SNM-S to retrieve the next set of latest notification messages and await for response from SNM-S.

6.2.3.2.2 SNM server procedure

Upon reception of an HTTP GET request from SNM-C where the Request-URI of the HTTP POST request is set to the notification URL of the SNM-S, the SNM-S:

a) shall determine the requestor identity of the received HTTP GET request as specified in clause 6.2.1.1, and:

1) if the identity of the sender of the received HTTP GET request is not authorized user, shall respond with an HTTP 403 (Forbidden) response to the HTTP GET request and skip rest of the steps;

b) shall determine whether notification URL corresponding to the one issued by SNM-S as part of create notification channel response or not; and:

1) if notification URL is not valid, shall respond with an HTTP 404 (Not Found) response to the HTTP GET request and skip rest of the steps;

c) shall determine whether notification channel corresponding to the channel identifier and exists or not; and:

1) if notification channel does not exist, shall respond with an HTTP 406 (Not Acceptable) response to the HTTP GET request and skip rest of the steps;

d) shall generate the pull notification message response to send notification messages received from the VAL servers:

1) shall include a Content-Type header field set to "application/vnd.3gpp.seal-notification-payload/json";

2) shall generate the notification message payload as specified in clause A.2.2:

i) shall set the channel identifier associated with the SNM-C; and

ii) shall generate the notification message list for the messaged received from VAL servers as specified in clause A.2.2-2; and

3) shall include an HTTP entity-body with the parameters specified in clause A.2.2 serialized into a JavaScript Object Notation (JSON) structure as specified in IETF RFC 7159 [5]; and

e) shall send an HTTP 200 (OK) response towards SNM-C.

NOTE: Efficient utilization of same TCP connection across multiple pull requests using keep-alive and other such HTTP feature is implementation specific.

6.2.4 Notification channel deletion procedures

6.2.4.1 SNM client procedures

Upon receiving a request from VAL service to stop receiving notifications via the notification channel; the SNM-C shall delete a notification channel by sending an HTTP DELETE request to the SNM-S. In the HTTP DELETE request the SNM-C:

a) shall set the Request-URI to the URI of the SNM-S;

b) shall include the Host header with public user identity of SNM-S;

c) shall include an Authorization header field with the "Bearer" authentication scheme set to an access token of the "bearer" token type as specified in IETF RFC 6750 [4];

d) shall include a Content-Type header field set to "application/vnd.3gpp.seal-delete-notification-channel-request";

e) shall generate the delete notification channel request message as specified in clause A.3.2:

   1) shall set the requestor identity to the notification management client identity; and

   2) shall set the channel identifier to the identity of the corresponding notification channel with SNM-S to be deleted. If the other VAL services use the same notification channel in the UE, then SNM-C:

      i) may set the VAL identity cluster info parameter with VAL identities corresponding to the VAL service requesting to stop receiving notifications, which indicates SNM-C prefers to deregister the notification channel for these identities rather than deletion of the notification channel; and

f) include the parameters specified in clause A.3.2 serialized into a JavaScript Object Notation (JSON) structure as specified in IETF RFC 7159 [5].

6.2.4.2 SNM server procedures

Upon reception of an HTTP DELETE request from SNM-C where the Request-URI of the HTTP DELETE request contains the URI of the SNM-S, the SNM-S:

a) shall determine the requestor identity of the received HTTP DELETE request as specified in clause 6.2.1.1, and:

   1) if the identity of the sender of the received HTTP DELETE request is not authorized user, shall respond with an HTTP 403 (Forbidden) response to the HTTP DELETE request and skip rest of the steps;

b) shall determine whether notification channel corresponding to the channel identifier exists or not; and:

   1) if notification channel does not exist, shall respond with an HTTP 406 (Not Acceptable) response to the HTTP DELETE request and skip rest of the steps;

c) shall delete the notification channel if the VAL identity cluster info parameter is not provided else proceed to only deregister the notification channel for these identities shared in VAL identity cluster info parameter corresponding to VAL service; and

d) shall send an HTTP 200 (OK) response to the SNM-C.

6.2.5 Notification channel update procedure

6.2.5.1 SNM client procedures

Upon detecting the expiry period of the active notification channel approaching; the SNM-C shall request for update of the notification channel expiry period by sending an HTTP PUT request to the SNM-S. In the HTTP PUT request the SNM-C:

a) shall set the Request-URI to the URI of the SNM-S;

b) shall include the Host header with public user identity of SNM-S;

c) shall include an Authorization header field with the "Bearer" authentication scheme set to an access token of the "bearer" token type as specified in IETF RFC 6750 [4];

d) shall include a Content-Type header field set to "application/vnd.3gpp.seal-update-notification-channel-request";

e) shall generate the update notification channel request message as specified in clause A.4.2:

   1) shall set the requestor identity to the notification management client identity;

   2) shall set the channel identifier to the identity of the corresponding notification channel with SNM-S to be updated; and

   3) may set the expiry time of the notification channel; and

f) include the parameters specified in clause A.4.2 serialized into a JavaScript Object Notation (JSON) structure as specified in IETF RFC 7159 [5].

Upon receiving an HTTP 200 (OK), the SNM-C shall store the expiry time received in the response for performing the future update procedure.

6.2.5.2 SNM server procedures

Upon reception of an HTTP PUT request from SNM-C where the Request-URI of the HTTP PUT request contains the URI of the SNM-S, the SNM-S:

a) shall determine the requestor identity of the received HTTP PUT request as specified in clause 6.2.1.1, and:

   1) if the identity of the sender of the received HTTP PUT request is not authorized user, shall respond with an HTTP 403 (Forbidden) response to the HTTP PUT request and skip rest of the steps;

b) shall determine whether notification channel corresponding to the channel identifier exists or not; and:

   1) if notification channel does not exist, shall respond with an HTTP 406 (Not Acceptable) response to the HTTP PUT request and skip rest of the steps;

c) shall generate a notification channel update response message with below attributes as specified in clause A.4.3;

   1) shall check whether the new proposed expiry time from SNM-C is valid or generate the new validity duration of the notification channel; and

d) include the parameters specified in clause A.4.3 serialized into a JavaScript Object Notation (JSON) structure as specified in IETF RFC 7159 [5]

e) shall include a Content-Type header field set to "application/vnd.3gpp.seal-update-notification-channel-response"; and

f) shall send an HTTP 200 (OK) response to the SNM-C.

6.3 Off-network procedures

Editor’s note: This clause will describe the off-network procedures based on 3GPP TS 23.434 [2].

Annex A (normative): Parameters for different operations

A.1 Creating notification channel

A.1.1 General

The information in this annex provides a normative description of the parameters which will be sent by SNM-C while creating notification channel request and the parameters which will be sent by SNM-S as a response to request for creating notification channel.

A.1.2 Client side parameters

The SNM-C shall convey the following parameters while sending request for create notification channel request.

Table A.1.2-1: Client side parameters for create notification channel request

Parameter	Description
Requestor identity	REQUIRED. Represents the identity of the notification management client.
Channel type	REQUIRED. Represents PULL or PUSH method to be used for delivering the notification messages. - 0x01: PUSH TYPE - 0x02: PULL TYPE
PUSH channel details	OPTIONAL. Represents details of the type of PUSH delivery and its associated data as specified in table A.1.2-2.
Expiry Time	REQUIRED. Represents the duration the notification channel shall be active (i.e. channel lifetime) as requested by the notification management client.
Pull Notification Message Trigger	OPTIONAL. Represents the application trigger for pulling notification messages. When set to "true", it indicates to the SNM-S about the SNM-C capability to support application trigger to initiate pull notification message procedure. Set to "false" or omitted otherwise. (NOTE)
VAL Id Cluster List	REQUIRED. Represents the list of VAL identities corresponding to the VAL services as specified in table A.1.2-3.
NOTE: This attribute may be provided if "Channel type" is set to PULL TYPE.	

Table A.1.2-2: PUSH channel details information

Parameter	Description
PUSH Callback-URL	REQUIRED. Represents PUSH call back URL where the SNM-S shall push the notification messages received by the VAL server.

Table A.1.2-3: VAL Identity Cluster Info

Parameter	Description
VAL User Identity	REQUIRED. Represents the identity of VAL user within the VAL service.
VAL Service ID	REQUIRED. Represents the Identify of the VAL service.
VAL Application ID	REQUIRED. Represents the Identity of the VAL application residing in the VAL UE.

A.1.3 Server side parameters

The SNM-S shall convey the following parameters while sending response to the create notification channel request.

Table A.1.3-1: Server side parameters for create notification channel response

Parameter	Description
Notification URL	OPTIONAL. Represents the URL that shall be used by SNM-C to pull notification if the channel type is PULL.
Callback URL	REQUIRED. Represents the URL, which shall be notified to VAL client by SNM-C. Further this URL shall be shared by VAL client to VAL server while subscribing for a VAL services.
Channel Identifier	REQUIRED. Represents the identifier of the newly created notification channel.
Pull Notification Message Trigger	OPTIONAL. Represents the application trigger for pulling notification messages. When set to "true", it indicates the SNM-S is capable to send application trigger to SNM-C to initiate pull notification message procedure. Set to "false" or omitted otherwise. (NOTE)
Expiry Time	OPTIONAL. Represents the duration the notification channel shall be active (i.e. channel lifetime) as requested by the notification management client.
NOTE: This attribute may be provided for PULL notification channel type.	

A.2 Receive notification messages

A.2.1 General

The information in this annex provides a normative description of the parameters which will be sent by SNM-S to the SNM-C, while sending notification messages over notification channel.

A.2.2 Server side parameters

The SNM-S shall convey the following parameters while sending the notification messages over notification channel.

Table A.2.2-1: Server side parameters for notification message payload

Parameter	Description
Channel Identifier	REQUIRED. Represents the identifier of the notification channel corresponding to the SNM-C.
VAL Notification Message List	REQUIRED. Represents a list of notification messages. Each notification message represents the message received from VAL servers which is encoded by SNM-S as specified in table A.2.2-2.

Table A.2.2-2: VAL Notification Message

Parameter	Description
VAL Id Cluster Info	REQUIRED. Represents the VAL identities shared by VAL server along with the VAL notification message which is encoded as specified in table A.1.2-3.
VAL Notification Message Type	REQUIRED. Represents the content type of the VAL notification message as shared by the VAL server.
VAL Notification Message Length	REQUIRED. Represents the length of the VAL notification message.
VAL Notification Message	REQUIRED. Represents the message received from the VAL server to be notified to the SNM-C.

A.2.3 Client side parameters

The SNM-C shall convey the following parameters while pulling the notification messages over notification channel.

Table A.2.3-1: Client side parameters for pull notification message

Parameter	Description
Requestor Identity	REQUIRED. Represents the identity of the notification management client.
Channel Identifier	REQUIRED. Represents the identifier of the notification channel corresponding to the SNM-C.

A.3 Delete notification channel

A.3.1 General

The information in this annex provides a normative description of the parameters which will be sent by SNM-C to the SNM-S as part of delete notification channel request over notification channel.

A.3.2 Client side parameters

The SNM-C shall convey the following parameters while sending request for delete notification channel request.

Table A.3.2-1: Client side parameters for delete notification channel request

Parameter	Description
Requestor identity	REQUIRED. Represents the identity of the notification management client.
Channel Identifier	REQUIRED. Represents the identifier of the notification channel corresponding to the SNM-C to be deleted.
VAL Identity Cluster Info	OPTIONAL. Represents the VAL identities to be deregistered from the notification channel encoded as specified in table A.1.2-3.

A.4 Update notification channel

A.4.1 General

The information in this annex provides a normative description of the parameters which will be sent by SNM-C to the SNM-S as part of update notification channel request over notification channel and the parameters which will be sent by SNM-S as a response to update notification channel request.

A.4.2 Client side parameters

The SNM-C shall convey the following parameters while sending request for update notification channel request.

Table A.4.2-1: Client side parameters for update notification channel request

Parameter	Description
Requestor identity	REQUIRED. Represents the identity of the notification management client.
Channel Identifier	REQUIRED. Represents the identifier of the notification channel corresponding to the SNM-C to be updated.
Expiry Time	OPTIONAL. Represents the duration the notification channel shall be active.

A.4.3 Server side parameters

The SNM-S shall convey the following parameters while sending response to the update notification channel request.

Table A.4.3-1: Server side parameters for update notification channel response

Parameter	Description
Expiry Time	OPTIONAL. Represents the duration the notification channel shall be active (i.e. channel lifetime) as indicated by the notification management server.

Annex B (informative): Change history

Change history							
Date	**Meeting**	**TDoc**	**CR**	**Rev**	**Cat**	**Subject/Comment**	**New version**
2023-02	CT1#140	C1-230655				Draft skeleton provided by the rapporteur.	0.0.0
2023-03	CT1#140					Implementing the following p-CR agreed in CT1: C1-230538, C1-230544, C1-230572, C1-230873	0.1.0
2023-04	CT1#141-e					Implementing the following p-CR agreed in CT1: C1-232796, C1-232797, C1-232798	0.2.0
2023-05	CT1#142					Implementing the following p-CR agreed in CT1: C1-233924, C1-233925, C1-233926, C1-233927	0.3.0
2023-08	CT1#143					Implementing the following p-CR agreed in CT1: C1-235408, C1-236029, C1-236030	0.4.0
2023-09	CT#101					Version 1.0.0 created for CT Plenary for information.	1.0.0
2023-11	CT#145	C1-239392				Pseudo-CR to share support for application triggers to PULL notification messages.	1.1.0
2024-03	CT#103	CP-240252				Presentation to TSG CT#103 for approval	2.0.0
2024-03	CT#103					Approved in CT#103	18.0.0
2024-06	CT#103					Correction of the change history table. Other contents remain the same as version 18.0.0.	18.0.1

History

Document history		
V18.0.1	June 2024	Publication