# ETSI GS PDL 023 V1.1.1 (2024-04)


# PDL service enablers for Decentralized Identification and Trust Management

***Disclaimer***

The present document has been produced and approved by the Permissioned Distributed Ledger (PDL) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.
It does not necessarily represent the views of the entire ETSI membership.

Reference
DGS/PDL-0023_DID_Framework

Keywords
decentralized identifier, PDL

***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:
https://www.etsi.org/standards-search

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 at www.etsi.org/deliver.

Users of the present document should be aware that the document may be subject to revision or change of status.
Information on the current status of this and other ETSI documents is available at
https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

If you find a security vulnerability in the present document, please report it through our
Coordinated Vulnerability Disclosure Program:
https://www.etsi.org/standards/coordinated-vulnerability-disclosure

***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 2024.
All rights reserved.

# Contents

# 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.

# Foreword

This Group Specification (GS) has been produced by ETSI Industry Specification Group (ISG) Permissioned Distributed Ledger (PDL).

# 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.

# 1 Scope

The present document defines Permissioned Distributed Ledger (PDL) Platform services to enable a decentralized identification and trust management framework. The present document also describes the characteristics and behaviour of this framework, along with the services that it offers and ideal solutions that can be built using it.

The objective of the present document includes:

- To define PDL platform services to handle registration management of different type of entities/participants to operate over the PDL platform to accomplish their specific tasks and purpose to realize the overall decentralized identification and trust management process.
- To define PDL platform services to handle decentralized identifier(s), related documents, and verifiable credentials.
- To define PDL platform services to verify the decentralized identifier and related information to enable a specific service provision (e.g. public, and private services).

In scope:

- Definition of Functionalities, Interface Reference points, and Procedures.

The approach taken in the present document is to focus on defining what needs to happen, not how it is implemented.

# 2 References

## 2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

[1] ETSI GS PDL 012 (V1.1.1): "Permissioned Distributed Ledger (PDL); Reference Architecture".

## 2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] W3C Recommendation 19 July 2022: "Decentralized Identifiers (DIDs) v1.0 - Core architecture, data model, and representations".

[i.2] ETSI GR PDL 019 (V1.1.1): "PDL Services for Decentralized Identity and Trust Management".

[i.3] EIDAS: "Supported Self-Sovereign Identity", May 2019.

[i.4] ENISA: "Digital Identity, Leveraging the Self-Sovereign Identity (SSI) Concept to Build Trust", January 2022.

# 3 Definition of terms, symbols and abbreviations

## 3.1 Terms

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

**applications at end-device:** application (e.g. a client application or wallet) used by the DID holder or Controller to generate, manage, store, or use private and public key pairs for related security (e.g. confidentiality and/or integrity protection)

NOTE: The sensitive information (such as cryptographic materials) may need to be protected by the "secure element" within the device or wallet. In such as case, the use of the cryptographic key(s) is restricted to the DID holder or controller respectively.

**Decentralized IDentifier (DID):** digital identifier managed through decentralized platform where the subject (e.g. end-user/device/any entity) possess the full control over its generation and associated data exposure, independent from any centralized registry, identity provider, or certificate authority

NOTE: DIDs can be URLs/URIs that relate a DID subject which enables trustable interactions with that subject. DID can refer to any subject (e.g. a person, organization, thing, data model, abstract entity, etc.) as determined by the controller of the DID. A DID is considered as a form of pseudonym as used in eIDAS and it is not directly linked to a formal identifier of the natural or legal person.

**DID controller:** controller of a DID is the entity (person, organization, or autonomous software) that has the capability as defined by a DID method and indicated in the DID document to make any changes to a DID document

NOTE: A DID holder can be the DID controller in some cases (or) a DID controller can be a different entity as authorized by the DID holder. DID controller holds the proof of possession or control of the holder's private key and will be responsible for issuance of a unique and anonymous DID to the holder.

**DID document:** DIDs resolve to the DID Documents, i.e. a set of simple documents that contains information associated to a DID (e.g. to verify the DID) and describes how to use a specific DID

NOTE: Each DID Document may contain at least three information such as proof purposes, verification methods (such as cryptographic public keys), and service endpoints (can also indicate services relevant to interactions with the DID holder). Proof purposes are combined with verification methods to enable mechanisms for proving various aspects related to DID holder's identification, authentication, and authorization.

**DID holder:** subject which is referred by the DID is called as the DID holder

NOTE: A DID holder in some cases can generate the DID and, in such cases, the DID holder is also referred as a DID controller.

**DID resolver function:** DIDs can be resolvable to their corresponding DID documents, where a DID Resolver function supports storage of DIDs and returns necessary data to access DID documents

**DID verification:** allows authentication of the subject identified by the DID

NOTE: The DID holder presents the data derived from one or more VCs, issued by one or more VC issuers, with a specific verifier (i.e. a service provider) to request and receive specific service of interest to the DID holder. A verifiable presentation is a tamper resistant/evident presentation encoded (with cryptographic methods) in such a way that authorship of the data can be trusted after a process of cryptographic verification. The DID holder authentication is facilitated with protocol exchanges between the DID holder, DID verifier and the trust management framework to verify the DID and validate the VCs (as part of authentication) to check if that can be sufficient to provide a requested service (i.e. resource access) for the DID holder.

**DID verifier:** it is a role that any third-party service provider or application server would perform to identify and authenticate the DID holder using the trust management framework

**distributed ledgers:** record of data stored by consensus with cryptographic audit trail maintained and validated by nodes in a decentralized platform based on governance

NOTE: Distributed ledger can be of two types such as permissioned distributed ledger and permission less distributed ledger. As the Permissioned Distributed Ledger (PDL) is further used in the present document, PDL service is further clarified below.

**off-chain storage:** privacy sensitive data associated to the DID can be stored and managed in isolation using off-chain methods or using any local/external authorized storage space as required

**PDL services:** it can facilitate the storage of DID related data such as DID documents, VC, etc.

NOTE: The ledgers which store the DID related data should be considered as a form of secure area (e.g. secure element or trusted platform). For example., the storage of DID can be supported through use of an agent service (such as PDL platform service if a distributed ledger is implemented for the storage) to remotely access the data from the end device and controlled through multiple authentication and authorization factors.

**Verifiable Credentials (VC):** are tamper-evident credentials that has authorship which can be cryptographically verified, and it includes one or more claims asserted by the VC issuer for the DID holder (i.e. subject)

NOTE: In practice, DIDs are used in combination with VCs to enable trusted digital interactions, where the required information about the subject is shared with third parties, by proving to those third parties that the DID subject has ownership of certain attestations or attributes. This proof is based on the cryptographic link between the VC, the DID subject the VC is about, and the issuer of the VC, which can be the DID subject itself (self-asserted claims), or another trusted entity.

**VC Issuer:** is an entity (e.g. a trusted entity or a trust service provider) that performs claims assertation about one or more subjects, creates a VC from the claims, and transmits the VC to the holder

NOTE: Trust on the VC is established either by trusting the issuer's DID (e.g. by out-of-band mechanisms, bilateral relationship, trusted lists etc,) or by any other means. The third party (e.g. service provider) can then use the presented cryptographically protected proof (i.e. the VC) to verify the ownership and trustworthiness of the claims about the subject.

**VC Storage:** to enable usage of VCs, the system that implements VC storage performs mediation service for the creation and verification of the identifiers, keys, and other relevant data, such as VC schemas, revocation of VC data, issuer public keys, and so on, e.g. some configurations may require correlation of identifiers for subjects

## 3.2 Symbols

Void.

## 3.3 Abbreviations

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

| | |
|---|---|
| DDRS | DID Document Registry Service |
| DID | Decentralized IDentifier |
| DLT | Distributed Ledger Technology |
| DPRS | DID operational Participants Registry service |
| DRS | DID Resolver Service |
| DTMF | Decentralized identification and Trust Management Framework |
| DVMS | DID Verification Management Service |
| eIDAS | electronic IDentification, Authentication, and trust Services |
| ID | IDentifier |
| L-RMS | Ledger Role-based registration Management Service |
| L-RMS-ID | Ledger Role-based registration Management Service IDentifier |
| NWK-ID | NetWorK IDentifier |

| | |
|---|---|
| PDL | Permissioned Distributed Ledger |
| RMS | Registration Management Service |
| SLA | Service Level Agreement |
| SSI | Self-Sovereign Identity |
| URI | Uniform Resource Identifier |
| URL | Uniform Resource Locator |
| VC | Verifiable Credentials |
| VDRS | Verifiable credentials Data Registry Service |
| W3C | World Wide Web Consortium |

# 4 Introduction (informative)

With the evolution of technologies, businesses, and advanced services, a more seamless, user friendly, user controlled, and privacy preserved identity management system is most essential for the quick roll-out and success of any business and services. Meanwhile, the trust in the identity of the subject or object (i.e. a natural or legal person, entity, etc.) has become the cornerstone of all digital services and activities. Here comes the decentralized identification, where a decentralized identifier is considered as the most suitable candidate which can link various essential and limited set of attributes (specific to the end-user(s) or device) as required for any specific service that can be shared with the service provider(s) or verifier(s) in order to enable authentication of the end-user/device to offer a specific set of service(s). The present document defines various PDL platform services such as role-based registration management, DID operation participants registry service, DID registry service, DID Resolver service, DID document registry service, Verifiable credentials data registry service, and DID verification management service, to enable the overall decentralized identification and trust management process. All forms of decentralized identification methodologies can utilize the PDL platform services defined in the present document to handle related data and trust management over the PDL platform. Specific implementation details (e.g. Implementation of identity using a specific method) are out of scope of the present document.

# 5 ETSI-ISG-PDL Decentralized Identification and Trust management Framework

## 5.1 Definition of terminologies

A decentralized identification and trust management process enables authentication of the end-user(s)/device(s) (i.e. to set up the initial trust between the end-user/device being the service consumer and the service provider). The key enablers to realize a fully functional decentralized identification and trust management involves various operational aspects such as DID generation (i.e. by a DID holder or DID controller), VC issuance (by a VC Issuer), DID storage and management, VC storage and management, Verification of the DID (by a DID verifier) for identification and authentication as listed and described in detail below.

## 5.2 Reference Framework Overview

### 5.2.1 Introduction

The present document uses a functional block architecture to define various services required to enable a decentralized identification and trust management framework. A decentralized platform has the capability to facilitate a globally unique digital identifier (i.e. DID with no possibility of duplication) related data management and control of associated cryptographic verification data, service information, etc. as needed for decentralized identification and authentication of a DID holder (i.e. user/device) to setup trusted interactions between the DID holder and a service provider for any related digital service provisions. The procedural aspects of PDL based decentralized identification and trust management ranges from different participants registration along with access control over the decentralized identification system, related data storage and management operations (e.g. throughout the data lifecycle), the decentralized identifier verification, and selective data exposure to service provider(s) for end-user/device authentication respectively.

A Decentralized Identification and Trust management framework can utilize the PDL services described in the PDL reference architecture (ETSI GS PDL 012 [1]) for the governance related aspects and the decentralized identification management and operation specific PDL services as shown in Figure 5.2.1-1. The core PDL service functionalities (i.e. capabilities, behaviour, and relationships) which forms the building block of decentralized identification and trust management process includes the following as shown in Figure 5.2.1-1 and it is described in detail in clause 5.2.2:

- Ledger Role-based registration management service;
- DID Operational participants Registry service;
- DID Resolver service;
- DID Document Registry service;
- VC Data Registry service; and
- DID Verification management service.

[R1] An ETSI-ISG-PDL compliant PDL platform SHALL include Mandatory Services required by the applications using a decentralized Identification and trust management framework.

[O1] An ETSI-ISG-PDL compliant PDL platform MAY include one or more of Optional Services required by the applications using such platform.

[R2] An ETSI-ISG-PDL compliant PDL platform SHALL include registry service(s) to manage the registered participant specific data such as DID, DID documents and VC Data.

[O2] An ETSI-ISG-PDL compliant PDL platform MAY use a unified registry service for storing DID, Documents and VC data specific to the DID Holder.

[O3] An ETSI-ISG-PDL compliant PDL platform MAY use dedicated registry service for storing different data types such as DID with DID Documents and VC data specific to the DID Holder.

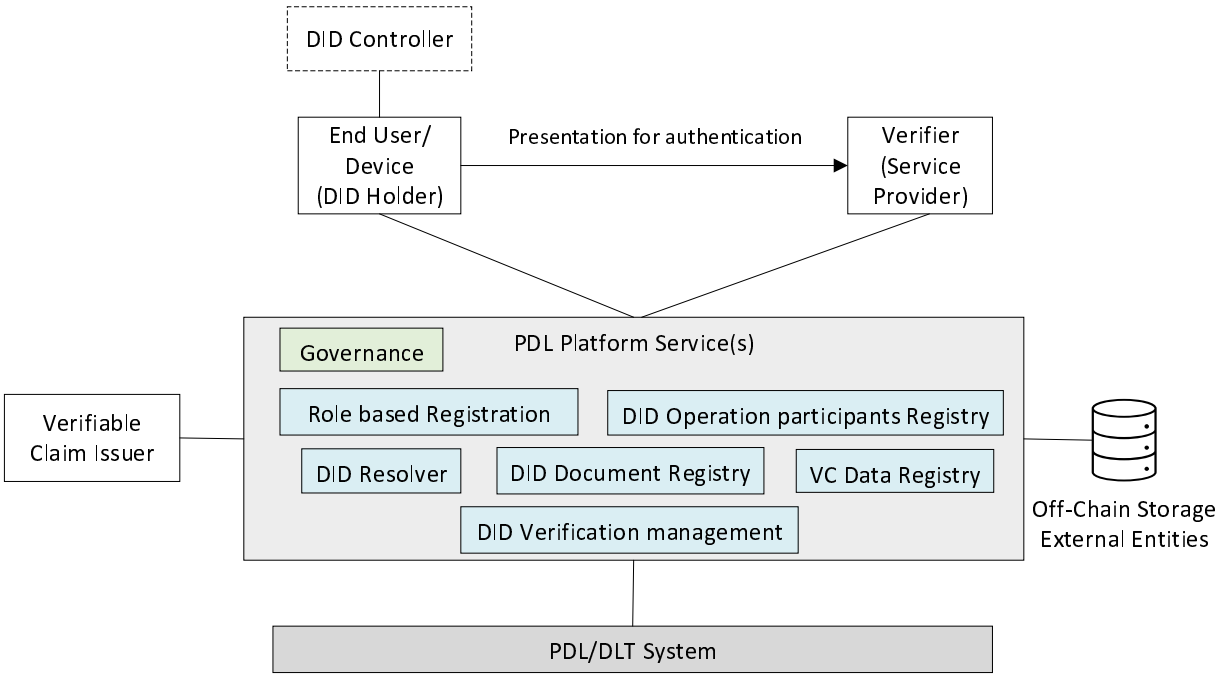


**Figure 5.2.1-1: PDL based Decentralized Identification and Trust management framework**

### 5.2.2 PDL services for decentralized identification and trust management

#### 5.2.2.1 Ledger Role-based registration management service

The Ledger role-based Registration Management Service (L-RMS) considers the following different roles for the participants who are the integral users of the decentralized identification and trust management framework. It provides registration service (along with authorization for fine grained access control) specific to the corresponding roles of the participants and their allowed operations in the PDL platform. The role-based registration management service offers registration and de-registration (e.g. revocation of registration) related services for different participants:

- Identity Holder (i.e. a DID Holder);
- Identity Controller (i.e. a DID Controller);
- VC Issuer; and
- DID Verifier.

[O2] An ETSI-ISG-PDL compliant PDL platform MAY include Decentralized Identification and Trust management framework related functionalities.

If the PDL platform supports Decentralized identification and Trust Management Framework (DTMF) related functionalities and operations, then further requirement(s) described in this clause are applicable.

[R2] An ETSI-ISG-PDL compliant PDL platform SHALL include Ledger Role-based registration management service to manage the registration and operation of different participants (entities) utilizing such a framework.

#### 5.2.2.2 DID Operational participants Registry service

The DID operational Participants Registry Service (DPRS) records and keeps track of the registered and de-registered participants from the PDL platform based DTMF by considering the instructions from the L-RMS.

[R3] An ETSI-ISG-PDL compliant PDL platform SHALL include a registry service to record the registration and operational details of different participants (entities) utilizing such a framework.

#### 5.2.2.3 DID Resolver service

The DID Resolver Service (DRS) stores the DIDs in a DID registry, keeps track of the DID(s) and its associated DID document location information (e.g. address) to enable DID document fetching and verification by the authorized services and entities (i.e. DID Verifiers e.g. service providers).

#### 5.2.2.4 DID Document Registry service

The DID Document Registry Service (DDRS) allows to store and manage the DID documents associated to the DID to facilitate DID verification. Whereas each DID Document can contain at least three things: proof purposes, service specific information for which the DID document can be used, verification methods, and service endpoints. Proof purposes are combined with verification methods to provide mechanisms for proving things.

EXAMPLE: A DID Document can specify that a particular verification method, such as a cryptographic public key or pseudonymous biometric protocol, can be used to verify a proof that was created for the purpose of authentication.

Service endpoints enable trusted interactions with the DID holder as well as authorized verifier. The DID Document Registry service offers Create (i.e. to store), Update, Revoke DID documents (i.e. deletion) related service operations.

#### 5.2.2.5 VC Data Registry service

The VC Data Registry Service (VDRS) allows to store and manage the VCs associated to the DID to facilitate VC based DID verification and validation related to a service request. The VC Data Registry service offers Create, Update, and Revoke VCs related service operations.

#### 5.2.2.6 DID Verification management service

The DID Verification Management Service (DVMS) can be a composite service (ETSI GS PDL 012 [1]) that uses DID resolver service, DID document registry service, DID operation(al) participant registry service and VC Data Registry service to fetch necessary data related to verification of DID (i.e. authentication of the subject identified by the DID), and performs exposure of selective data to the verifier to enable authentication of the subject and authorization verification of subject to respective service(s).

## 5.3 PDL Framework Operations and Services

### 5.3.1 Registration Management

#### 5.3.1.1 Introduction

Registration Management procedure describes how any participant mentioned in clause 5.2.2.1 can register to the DTMF shown in Figure 5.2.1-1 to perform any of decentralized identification and trust management related operations using the PDL based DTMF. Further the registration management procedure also describes how a participant can be de-registered from the DTMF based on various conditions.

#### 5.3.1.2 Role-based registration

The detailed registration management procedure for different participants (taking different roles) is described in this clause as shown in Figure 5.3.1.2-1. The role-based registration procedure primarily involves two services such as Ledger Role-based registration management service and DID Operational participants Registry service described in clause 5.2.2. An entity (i.e. related to DID holder device, VC issuer and the DID Verifier) which requires to participate in the Decentralized identification and trust management process can use the registration procedure described in this clause to initially register to the DTMF to allow any further operations over the DTMF.

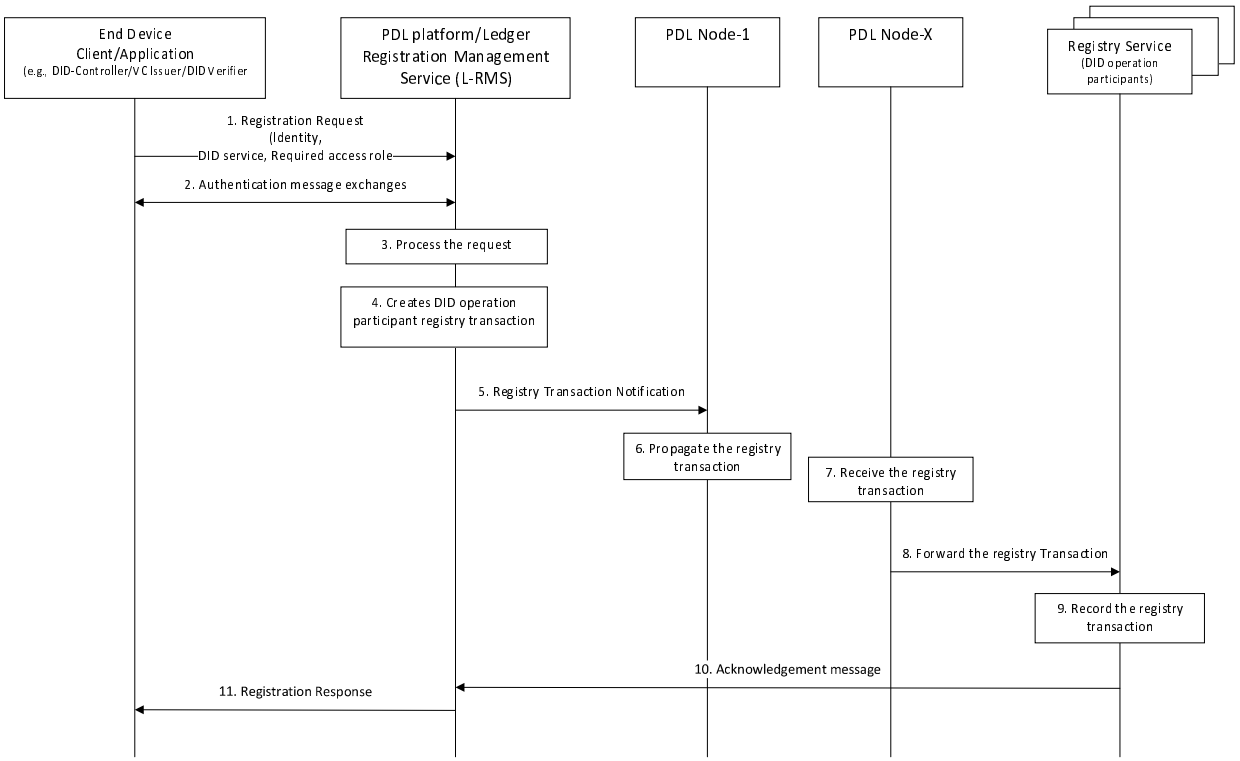


**Figure 5.3.1.2-1: Role-based registration Procedure**

If the DID holder wants to register to the DTMF, it performs the following steps:

1) The DID holder sends a registration request to the PDL platform's Ledger role-based Registration Management Service (L-RMS), which includes a source identity, service type information set as 'DID service' (i.e. to indicate that the registration is related to the DID related entity which need to act as the DID holder in the DTMF), required access role (indicates that DID holder access role is requested) and the actual DID (i.e. DID refers to decentralized identity/digital identity/Self-Sovereign Identity (SSI) [i.1], [i.2], [i.3] and [i.4]. It is generated either by the subject or by the subject controller in a privacy protected form to uniquely identify an entity e.g. DID Controller in case of internet of things).

NOTE 1: The entity may have registered to the PDL platform as a general user (e.g. using ETSI GS PDL 012 [1]) of the PDL platform, in which case the entity may have a source identity). In certain case of implementation if the entity has a client application or wallet installed to use the PDL platform, the access to the PDL platform can be authenticated using the identity and credentials (e.g. public-private key) associated to the client applications or the wallet.

[Conditional - applicable for DID Controller] In case, if a DID controller is involved in the registration procedure instead of DID holder, then in step 1 the DID controller sends to the L-RMS, a registration request with the required access role set as, 'DID Controller', Source Identity of the DID controller along with the other information described for step 1 above.

[Conditional - applicable for DID holder/Controller who also want to take the role of VC Issuer] In case, if a DID holder/controller is involved in the registration procedure, then in step 1 it sends to the L-RMS, a registration request with more than one required access role set as, 'DID holder/Controller as applicable', 'VC Issuer', along with the other information described for step 1 above.

[Conditional - applicable for VC Issuer] In case, if a VC Issuer is involved in the registration procedure, then in step 1 the VC Issuer sends to the L-RMS, a registration request where the required access role is set as, 'VC Issuer', and Source Identity of the VC Issuer is also included with the other information as described for step 1 above.

[Conditional - applicable for DID Verifier] In case, if a DID Verifier is involved in the registration procedure, then in step 1 the DID Verifier sends to the L-RMS a registration request where the required access role is set as, 'DID Verifier', and Source Identity of the DID Verifier is also included with the other information as described for step 1 above.

2) The L-RMS can initiate and perform mutual authentication (e.g. based on local policy) with the DID holder based on a preconfigured credentials (e.g. public-private key pair or any secret key associated to the client application or wallet).

3) On a successful mutual authentication, the L-RMS process the registration request.

4) The L-RMS determines to register the DID holder and it sets a registration ID for the DID holder. Further it creates a Registry transaction notification message which includes the L-RMS ID, target Registry service information (i.e. such as registry service name, and ID (e.g. address) related to the DID Operation(al) participant registry), Source Identity, Service type information (DID service), Registration ID, DID, Authorized access role (set as DID holder), Authorization code, and Lifetime (for the validity of the registration). Further the message can be transformed into a transaction (i.e. DID Operation(al) participant registry transaction) to add the new participant related registration information to the DID Operation(al) participant registry.

[Conditional - applicable for DID Controller] In case, if a DID controller is involved in the registration procedure instead of DID holder, then in step 4, the L-RMS determines to register the DID controller and it sets a registration ID for the DID controller. Further it creates Registry transaction notification message which includes the L-RMS ID, target Registry service information (i.e. such as registry service name, and ID (e.g. address) related to the DID Operation(al) participant registry), Source Identity, Service type information (DID service), Registration ID, DID, Authorized access role (set as DID controller), Authorization code, and Lifetime (for the validity of the registration). Further the message can be transformed into a transaction (i.e. DID Operation(al) participant registry transaction) to add the new participant to the DID Operation(al) participant registry).

[Conditional - applicable for VC Issuer] In case, if a VC Issuer is involved in the registration procedure, then in step 4, the L-RMS determines to register the VC Issuer and it sets a registration ID for the VC Issuer. Further it creates Registry transaction notification message which includes the L-RMS ID, target Registry service information (i.e. such as registry service name, and ID (e.g. address) related to the DID Operation(al) participant registry), Source Identity of the VC Issuer, Service type information (DID service), Registration ID, DID, Authorized access role (set as VC Issuer), Authorization code, and Lifetime (for the validity of the registration). Further the message can be transformed into a transaction (i.e. DID Operation(al) participant registry transaction) to add the new participant to the DID Operation(al) participant registry).

[Conditional - applicable for DID Verifier] In case, if a DID Verifier is involved in the registration procedure, then in step 4, the L-RMS determines to register the DID Verifier and so it sets a registration ID for the DID Verifier. Further it creates Registry transaction notification message which includes the L-RMS ID, target Registry service information (i.e. such as registry service name and ID (e.g. address) related to the DID Operation(al) participant registry), Source Identity of the DID Verifier, Service type information (DID service), Registration ID, DID, Authorized access role (set as DID Verifier), Authorization code, and Lifetime (for the validity of the registration). Further the message can be transformed into a transaction (i.e. DID Operation(al) participant registry transaction) to add the new participant to the DID Operation(al) participant registry).

5) The L-RMS sends to the configured PDL node a DID Operation(al) participant registry transaction (which includes the Registry transaction notification message).

NOTE 2: The message to transaction conversion is up to the PDL platform service provider implementation.

6) PDL Node-1 propagates the received transaction through the target PDL network.

7) PDL Node-X receives the transaction from the target PDL network as the result of transaction propagation.

8) After the transaction is validated and it is successfully stored to the ledger (e.g. as a result of PDL consensus process in a ledger related to the registry service associated to the DID Operation(al) participant registry). Also, the PDL Node-X forwards the transaction to the registry service based on the target Registry service information. The registry service transforms the transaction into message to recover the message (i.e. DID Operation(al) participant registry transaction notification message).

9) The registry service can store the DID Operation(al) participant registry transaction received as part of the Registry transaction notification message based on local policies, e.g. in a local storage/off-chain/ledger.

10) The registry service can send to L-RMS, an acknowledgement message with the L-RMS ID, Source ID, Registration ID, Registry service type ID, and the result as 'Success' indication.

11) The L-RMS sends to the end device, a Registration Response with L-RMS ID, Service type information (DID service), Registration ID, Authorized access role, Authorization information (e.g. code/token), and Lifetime.

Based on steps 1 to 2, if the L-RMS determines not to register the end device or application, it sends a Registration Response with 'failure' indication.

NOTE 3: The L-RMS can accept the required access role provided by the end-device/client/applicant (in step 1) based on the authentication results (e.g. based on end-user's information in the certificate or any SLA agreement which is outside the scope of the present document). So, based on the local policies, authentication credentials evaluation and authentication result the L-RMS can determine to agree or deny a required access role requested by the end-device/application.

NOTE 4: A smart contracts can be used by the registry services to keep track of lifetime related expirations, linking of all DID related entries, etc.

### 5.3.1.3 De-registration

The revocation of registration (related to a participant) from the DTMF is defined as the deregistration. The de-registration procedure for different participants (such as DID-Controller, VC Issuer, DID Verifier) is described in this clause as shown in Figure 5.3.1.3-1. The de-registration procedure primarily involves two services such as Ledger Role-based registration management service and DID Operational participants Registry service described in clause 5.2.2. The L-RMS or registry service (i.e. related to DID Operational participants) invokes de-registration of participant(s) from the DTMF based on the conditions listed below:

i) The registry service identifies that a participant's registration has expired based on the registration lifetime (e.g. based on local policy registration can be valid for, say, a month by default, and therefore the registration expires automatically when a month is lapsed).

NOTE 1: Based on DTMF implementation, the registration expiry can be identified by a smart contract or governance to assist the registry service for the same purpose.

ii) The registry service or L-RMS identifies that the registered participant has violated any allowed operations based on local policy.

NOTE 2: Based on DTMF implementation and local policy operational violations can be identified by a smart contract or governance to assist the L-RMS for the same purpose.

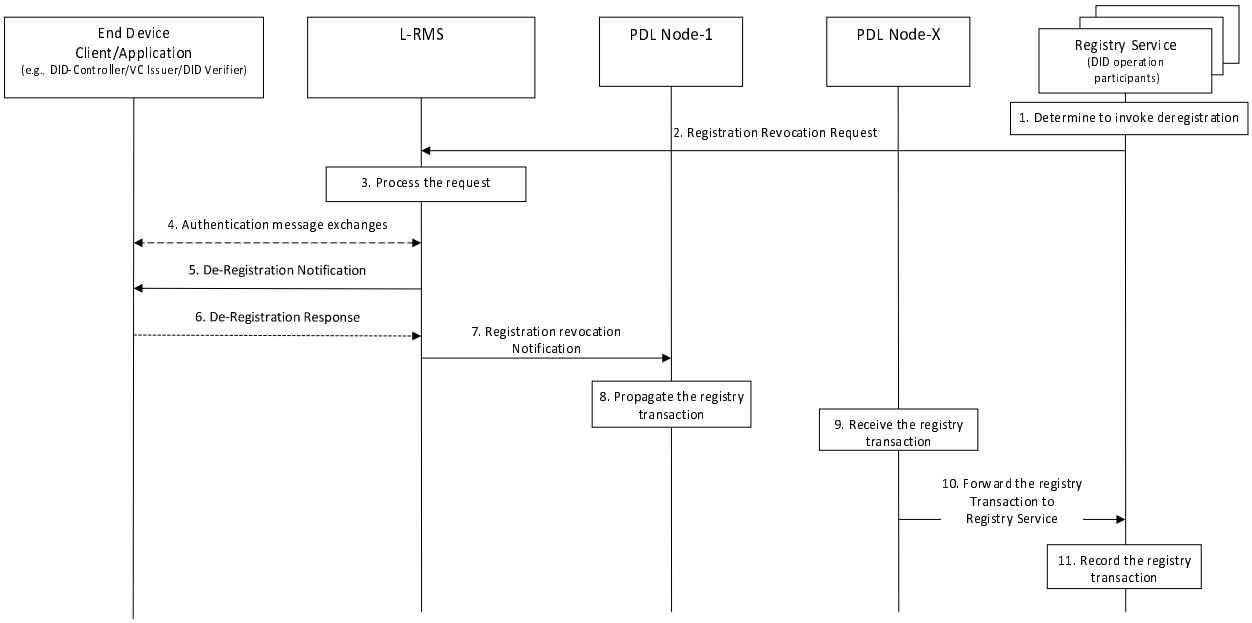


**Figure 5.3.1.3-1: De-registration Procedure**

If the L-RMS or Registry service determines to de-register a participant from the DTMF, it performs the following steps:

1) The DID operation participants registry service determines to revoke the registration for the registered participant(s).

2) The Registry service sends to the L-RMS, the Registration revocation request based on the L-RMS ID associated to the participant's Registration ID (i.e. available as part of DID Operation(al) participant registry information). The registration revocation request includes L-RMS ID, Registry service information (i.e. registry service name and ID (e.g. address) related to the DID Operation(al) participant registry)), Registration ID, authorized access role and a cause value related to the registration revocation (i.e. indicating lifetime expiry, error, any operational violation reason code).

3) The L-RMS on receiving the registration revocation request for a registered participant associated with its L-RMS ID, the L-RMS process and determine to invoke the registration revocation.

4) L-RMS can initiate and perform mutual authentication (e.g. based on local policy) with the target participant (e.g. can be DID-Controller/VC Issuer/DID Verifier) based on a preconfigured credentials (e.g. public-private key pair or any secret key associated to the client application or wallet).

   [Conditional] If the L-RMS determines based on local policy (as described in this clause) to revoke any registered participant, the L-RMS can perform steps 5 to 11 directly.

5) The L-RMS sends to the end-device, a de-registration notification message which includes the Service type information (DID service), Registration ID, and Cause value.

6) The end-device can send to the L-RMS, the de-registration response message with Registration ID and successful registration revocation acknowledgement indication. Further the end-device deletes all information associated to the registration such as registration ID, and authorization information.

7) The L-RMS creates a registration revocation notification message and coverts it to a transaction (i.e. registration revocation notification) and sends to the PDL Node-1 based on the local configuration. The registration revocation notification includes Registry service information (i.e. such as registry service name, and ID (e.g. address) related to the DID Operation(al) participant registry), L-RMS ID, Source Identity, Service type information (DID service), Registration ID, and Revoked Indication (e.g. revocation successful indication).

   If the L-RMS does not receive any de-registration response from the end-device in step 6, based on local policy (e.g. after a preconfigured waiting time), it performs step 7.

8) PDL Node-1 propagates the received transaction through the target PDL network.

9) PDL Node-X receives the transaction from the target PDL network as the result of transaction propagation.

10) After the transaction is validated, it is successfully stored to the ledger (e.g. as a result of PDL consensus process in a ledger related to the registry service) associated to the DID Operation(al) participant registry based on the target registry service type information. Also, the PDL Node-X forwards the transaction to the registry service and the registry service transforms the transaction into message to recover the message (i.e. registration revocation notification message).

11) The registry service stores the registration revocation notification based on local policies e.g. in the local storage/off-chain/ledger.

NOTE 3: Irrespective of the roles, any end-device/application related to an DID-Controller/VC Issuer/DID Verifier associated registration can be revoked using the procedure described in the Figure 5.3.1.3-1 by providing the corresponding access role information in step 2.

NOTE 4: Smart contracts can be configured to link and maintain the registration status (such as successful registration and respective revocations) related to a registration ID. Further the smart contracts can be used by the registry services to keep track of lifetime related expirations, linking of all DID related entries, etc.

## 5.3.2 Data Management

### 5.3.2.1 Introduction

Data Management procedure describes how an authorized participant (such as DID Holder/DID Controller and VC Issuer) mentioned in clause 5.2.2.1 can manage data (i.e. store, update or delete data) such as DIDs, DID documents, VCs over the DTMF as shown in Figure 5.3.2.2.1-1.

### 5.3.2.2 DID and DID Documents management

#### 5.3.2.2.1 Procedure

The management of DID and the related DID documents involves various operations such as listed below:

- Storage of DID and DID Documents (i.e. on request from the DID holder or DID controller).
- Update of DID and DID Documents (i.e. on request from the DID holder or DID controller).

- Deletion/Revocation of DID and DID Documents (i.e. on request from the DID holder/DID controller/Ledger-registration management service in the PDL platform).

The data management procedure for DID and DID documents primarily involve services such as DID Operational participants Registry service, DID Document Registry service, and DID Resolver service described in clause 5.2.2.

An entity (i.e. specific to DID holder/controller device associated to the DID holder) which requires to manage its DID and DID documents over DTMF use the data management procedure described in this clause as shown in Figure 5.3.2.2.1-1 for operations such as create, update and delete.

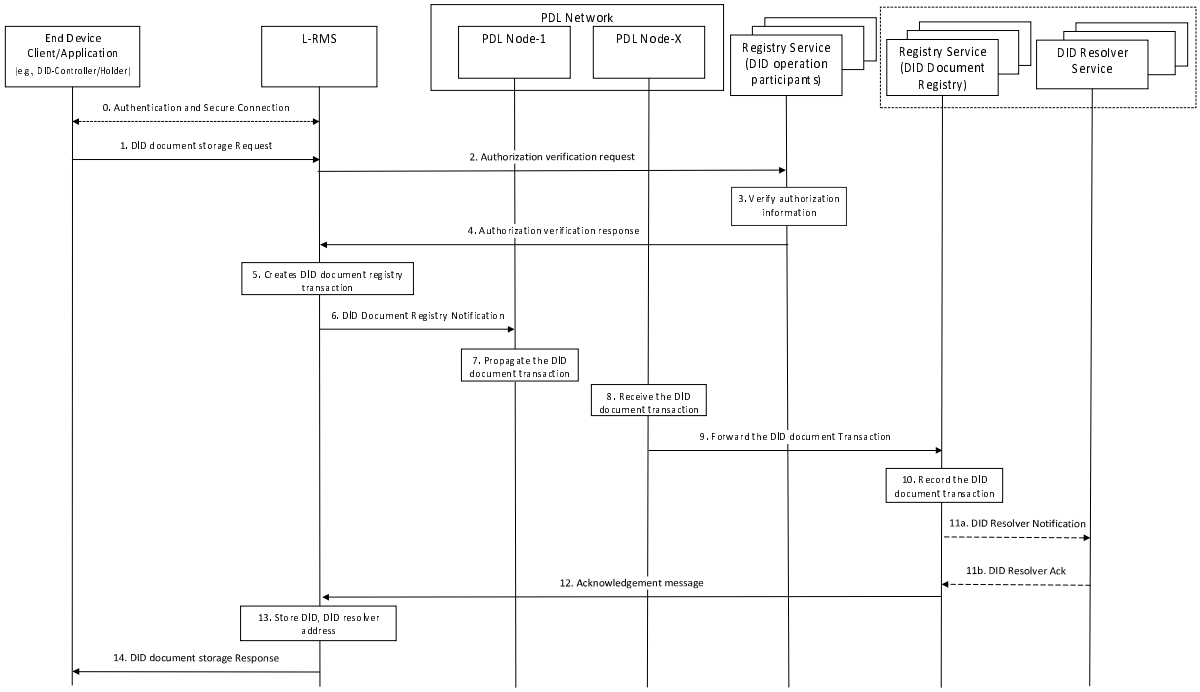


**Figure 5.3.2.2.1-1: DID and DID document management procedure**

## 5.3.2.2.2 DTMF Operations

### 5.3.2.2.2.1 DTMF - CREATE Operation

The DID holder or controller on generation of DID and DID documents uses the following procedure for DID and DID documents specific storage management, if it wants to use DTMF for storage management aspects i.e. specific to CREATE operation. The create operation helps to store a newly generated DID along with its DID documents:

NOTE 1: For simplicity the term DID holder is used to refer both the DID holder and the DID controller in the following procedure.

Precondition: If a secure connection exists (as shown in step 0 of Figure 5.3.2.2.1-1), the DID holder performs step 1. Else the DID holder and the L-RMS performs mutual authentication and sets up a secure connection before step 1, where the authentication can be based on Security Platform Services defined in ETSI GS PDL 012 [1].

1) The end device client/application send to the L-RMS, a DID document storage request with Registration ID, service type information (i.e. indicates DID service), access role (i.e. indicates DID holder/controller as authorized), authorization information (i.e. a code or token received as authorization information during a successful registration), the DID document(s), request type (set as Create).

NOTE 2: The end device indicates access role as 'DID holder' if it is the actual subject/end-user. Else if the end-device is related to a controller which manages the DID and DID documents on behalf of a subject/end-user, it indicates access role as DID controller.

NOTE 3: The DID document(s) can include information such as the DID, DID controller ID (if applicable/exists), verification method(s), cryptographic public key, service type/information, verifiable claims, URI related to claims, etc. Based on a method used for DID generation, the information in the DID document can vary and it can be specific to the implementation and outside the scope of the present document.

[Conditional- applicable for DID Controller] Figure 5.3.2.2.1-1, steps 1 to 14 can be applied with an additional adaptation that steps 1 to 12 also includes Source Identity (i.e. user identity e.g. a PDL user ID) corresponding to the DID holder (i.e. subject) in addition to the Registration ID of the DID Controller. Further, the access role information specific to the DID controller is used.

2) The L-RMS based on the local configuration and policies sends to the DID Operation(al) participant registry an authorization verification request message, with Registration ID, access role and authorization information.

3) The DID Operation(al) participant registry service verifies the authorization information and access role related to the Registration ID (e.g. by querying the respective ledger for a related transaction records or by checking an offline/local storage) to check if the access role, authorization information and registration ID matches with the records related to the registered participant.

4) If the authorization verification is successful in step 3, the DID Operation(al) participant registry service sends to the L-RMS, an authorization verification response message, with the registration ID and result as "successful" and proceed with step 5.

If the verification of registration ID, access role and authorization information do not match with the records, then an authorization verification response message, includes the registration ID, and result as "failure" and further step 14 failure case is executed.

5) The L-RMS generates a DID document registry notification message with the target Registry service information (i.e. registry service name, and ID (e.g. address) specific to the DID Document registry), request type (set as Create), L-RMS ID, Service type information (DID service), Registration ID, authorized access role, Lifetime, DID, and DID Document(s) (received in step 1).

Further the message is transformed into a DID document transaction to store the DID documents to the corresponding DID document registry.

NOTE 4: The message to transaction conversion is up to the PDL platform service provider implementation.

6) The L-RMS sends to the configured PDL node-1, the DID document transaction (which includes the DID document registry notification message).

NOTE 5: Based on different implementation, the L-RMS if it is capable to act as a PDL node, it can direct propagate the DID document transaction to the PDL network. In that case, step 7 is skipped.

7) PDL Node-1 propagates the received DID document transaction through the target PDL network.

NOTE 6: A group of PDL nodes say PDL 1, 2, and so on e.g. X (that participates in running the protocol software of a PDL network, it includes components such as computing resources e.g. bare metal or virtual machine on which the ledger resides, which also can include wallet, and application) to facilitate consensus, distributed storage, and management of ledger data.

8) PDL Node-X receives the DID document transaction from the target PDL network as the result of transaction propagation.

9) Once DID document transaction is validated, it is successfully stored to the ledger (e.g. as a result of PDL consensus process) associated to the DID Document. Also, the PDL Node-X forwards the DID document transaction to the DID Document registry based on the target Registry service information. The registry service transforms the transaction into message to recover the message (i.e. DID document registry notification message).

10) The DID document registry service stores the DID Document information received as part of the DID document Registry notification message based on local policies, e.g. in a local storage/off-chain.

11) The DID document registry service manages DID resolver services to facilitate DID based DID document fetching operation (e.g. for any 3rd party service):

    a) The DID document registry (based on local policy) sends to a DID resolver service, a DID resolver notification message with DID, request type (set as create) and DID Document Registry address (i.e. to fetch the DID documents.

    b) The DID resolver services stores the DID and the DID Document Registry address. Further the DID resolver service responds with a DID resolver Acknowledgement (Ack) message with DID and a success indication.

12) The DID Document registry service sends to the L-RMS, an acknowledgement message which includes the L-RMS ID (received in step 9 related to the DID document registry notification), Registration ID, DID Document Registry address, Success, and DID resolver registry service ID.

13) The L-RMS stores the DID, DID document registry address (as DID resolver information) and other received information locally or in off-chain.

14) The L-RMS sends to the end device client/application, a DID document storage response message with DID document registry address and Success.

    Alternatively, for the failure case described in step 4, the L-RMS sends to the end device client/application, a DID document storage response message with failure indication and a suitable cause value (i.e. such as violation code/authorization failure/authentication failure, etc.).

NOTE 7: The end device client/application while requesting service from any service provider, it can provide the DID document registry address together with DID to enable the service provider (i.e. who performs the role of DID verifier) to request the DID resolver for any DID documents to perform DID related authentication of the end-device, which is outside the scope of the present document.

NOTE 8: Based on different implementation, in Figure 5.3.2.2.1-1, alternatively the L-RMS can be a PDL node by itself and the L-RMS can propagate the PDL transaction (related to the DID document related data storage management notification) to the PDL network by itself for steps 6 to 8.

#### 5.3.2.2.2.2 DTMF - UPDATE Operation

The update operation helps to update the DID related DID documents and related information over the DTMF. The procedure described in clause 5.3.2.2.2.1 can be reused for the DID as well as DID document update operations with the following adaptations on request from the respective DID holder or DID controller:

1) In step 1, the end device client/application send to L-RMS, a DID document **storage** request with Registration ID, service type information (i.e. it indicates a DID service), access role (i.e. indicates DID holder/controller as authorized), authorization information (i.e. a code or token received as authorization information during a successful registration), the updated DID document(s), request type (set as Update).

2) In steps 5 to7, DID document registry notification message (as well as the related transaction) includes the **request type** 'set as Update' (instead of create indication) in addition to the other aspects.

3) In step 11a, the DID document registry send to a DID resolver service, a notification message with DID, **request type** 'set as Update' and DID Document Registry address.

4) In step 11b, the DID resolver services updates the DID and related DID Documents Registry address. Further the DID resolver service sends to the DID Document registry service, a DID resolver Acknowledgement (Ack) message which can include DID and a success indication.

#### 5.3.2.2.2.3 DTMF - DELETE Operation

The delete operation (i.e. revocation) helps to delete/revoke the DID related DID documents and related information from the DTMF. The procedure described in clause 5.3.2.2.2.1 can be reused for the DID as well as DID document delete operations with the following adaptations (i.e. on request from the respective DID holder or DID controller):

1) In step 1, the end device client/application can send to the PDL platform Ledger Role-based Registration Management Service (L-RMS), a DID document storage request with Registration ID, service type information (i.e. it indicates a DID service), access role (i.e. indicates DID holder/controller as authorized), and authorization information (i.e. a code or token received as authorization information during a successful registration) and request type (set as Delete).

2) In steps 5 to 7, DID document registry notification message (as well as the related transaction) includes the **request type** 'set as Delete' (instead of create indication) in addition to the other aspects.

3) In step 11a, the DID registry send to a DID resolver service, a DID resolver notification message with DID, **request type** 'set as Delete' and DID Document Registry address.

4) In step 11b: The DID resolver services **deletes** the DID related DID Document Registry address. Further the DID resolver service sends to the DID Document registry service, a DID resolver Acknowledgement (Ack) message which can include DID and a success indication.

NOTE 1: The smart contracts can be used by the registry services described in this clause to keep track of lifetime expirations and linking of all DID related entries etc. and it is up to the PDL network service operator implementation.

NOTE 2: The L-RMS based on local policy, if identifies that the registered participant has violated any allowed operations (as described in clause 5.3.1.3), it can directly perform step 5 (with DTMF- DELETE Operation specific adaptations) until step 11b.

### 5.3.2.3 Co-ordinated DID Document publishing to PDL

This clause describes how the DID holder and the PDL platform can co-ordinate to generate a DID document that will be published for a DID holder. Figure 5.3.2.3-1 illustrates a procedure of creating DID document for a DID Holder (e.g. a Device) and publishing it to a PDL Network via a PDL Node.

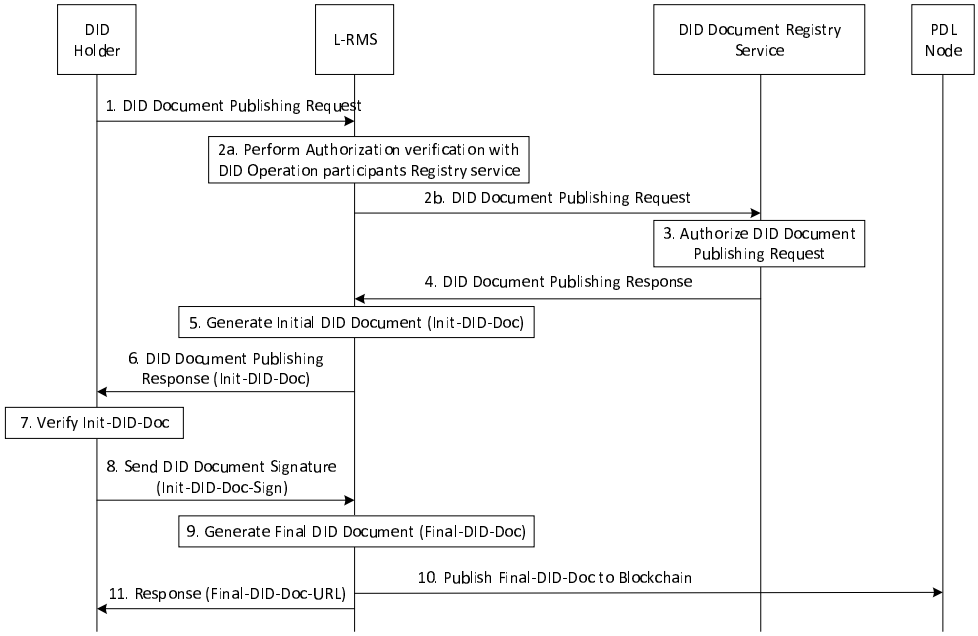


**Figure 5.3.2.3-1: Publish/Create DID Document for a DID Holder (e.g. a Device)**

1) The DID Holder sends a DID document publishing request to L-RMS. This request contains the following parameters in addition to the information described in clause 5.3.2.2.1 step 1:

   - DID: The DID of the DID Holder.
   - PDL-NWK-ID: The identifier of PDL Network where DID document will be published to.

2) a-b:

   a) The L-RMS performs authorization verification of the DID holder by using the DID Operation(al) participant registry service as described in clause 5.3.2.2.1 steps 2 to 4.

   b) Following the successful authorization verification, the L-RMS forwards the DID document publishing request to DID Document Registry Service.

3) DID Document Registry Service receives the request from step 2b. It checks its local policies to determine if the requested DID document publishing is allowed. If the request is approved, DID Document Registration Service determines signature authentication scheme for the DID Holder (DID-Holder-Sign-Auth-Scheme), which will be contained in the DID Document. DID Document Registration Service may re-determine PDL-NWK-ID.

4) DID Document Registration Service sends a response to L-RMS indicating if the request has been approved or rejected in step 3. If the request is approved, this response contains DID-Holder-Sign-Auth-Scheme, PDL-NWK-ID if it was re-determined in step 3, and the DID Holder's public key (DID-Holder-Public-Key).

5) L-RMS generates an initial DID Document for the DID Holder (Init-DID-Doc). The list of parameters to be included in Init-DID-Doc includes DID, DID-Holder-Sign-Auth-Scheme, DID-Holder-Public-Key, and L-RMS-ID (The identifier of L-RMS). L-RMS signs Init-DID-Doc using its private key (L-RMS-Private-Key) according to L-RMS's signature authentication scheme (L-RMS-Sign-Auth-Scheme); thus, Init-DID-Doc also contains L-RMS-Public-Key, L-RMS-Sign-Auth-Scheme and L-RMS's signature (L-RMS-DID-Doc-Sign).

6) L-RMS sends a DID Document publishing response to the DID Holder including Init-DID-Doc.

7) The DID Holder receives the response from step 6. It verifies Init-DID-Doc (i.e. to verify L-RMS-DID-Doc-Sign is a valid signature of L-RMS for Init-DID-Doc).

   If the signature verification in step 7 passed, the DID Holder uses its private key (DID-Holder-Private-Key) to generate its signature for Init-DID-Doc (i.e. DID-Holder-Init-DID-Doc-Sign). At this point, the DID Holder may update some parameters (e.g. introduce new parameters, update the value of existing parameters) as contained in Init-DID-Doc; then, DID-Holder-Init-DID-Doc-Sign needs be calculated considering new values of those updated parameters.

8) The DID Holder sends its signature DID-Holder-Init-DID-Doc-Sign to L-RMS. If the DID Holder updated any parameters in step 7, the DID Holder also sends the new values of those updated parameters to L-RMS via step 8.

9) L-RMS receives DID-Holder-Init-DID-Doc-Sign. L-RMS verifies DID-Holder-Init-DID-Doc-Sign using the DID Holder's public key (DID-Holder-Public-Key). If the verification of DID-Holder-Init-DID-Doc-Sign passes, L-RMS inserts it to Init-DID-Doc to generate the final DID Document (Final-DID-Doc).

   If step 8 indicated any changed parameters, L-RMS needs to re-generate Inti-DID-Doc to incorporate the changed parameters; then, L-RMS needs to verify the changes and re-generate L-RMS-DID-Doc-Sign. Then, L-RMS use the changed parameters and regenerated L-RMS-DID-Doc-Sign to update Init-DID-Doc in order to generate Final-DID-Doc.

10) L-RMS publishes Final-DID-Doc to PDL network as denoted by PDL-NWK-ID via PDL Node. L-RMS receives an address or URL for accessing the published Final-DID-Doc (i.e. Final-DID-Doc-URL) from PDL Node.

    Further following the successful publishing of the DID document, Clause 5.3.2.2.1. Steps 9 to 12 are performed by the DID document registry service to manage the DID and DID document registry service address information for the DID resolver service.

11) L-RMS sends a response to the DID Holder indicating Final-DID-Doc-URL and DID document registry address (as DID resolver information).

Alternatively, for the failure case described in step 4, the L-RMS sends to the end device client/application, a DID document storage response message with failure indication and a suitable cause value (i.e. such as violation code/authorization failure/authentication failure, etc.).

### 5.3.2.4 Verifiable Credentials management

#### 5.3.2.4.1 Procedure

This clause describes how the VC related data management is handled using DTMF services such as DID Operational participants Registry service, and VC Data Registry service described in clause 5.2.2.

The management of VC data involves various operations such as listed below:

- Storage of VC data (i.e. on request from the VC Issuer or DID holder/DID controller).
- Update of VC data (i.e. on request from the VC Issuer or DID holder/DID controller).
- Deletion/Revocation of DID and DID Documents (i.e. on request from the VC Issuer or DID holder/DID controller).

NOTE: The VC data specific to an DID holder can be issued by the VC Issuer. The Issued VC data can be stored over the DTMF on request from the entity specific to the VC Issuer or DID holder/Controller.

An entity (i.e. specific to VC Issuer or DID holder/controller device associated to the DID holder) which requires to manage the VC data over DTMF use the data management procedure described in this clause as shown in Figure 5.3.2.4.1-1 for operations such as create, update, and delete respectively.

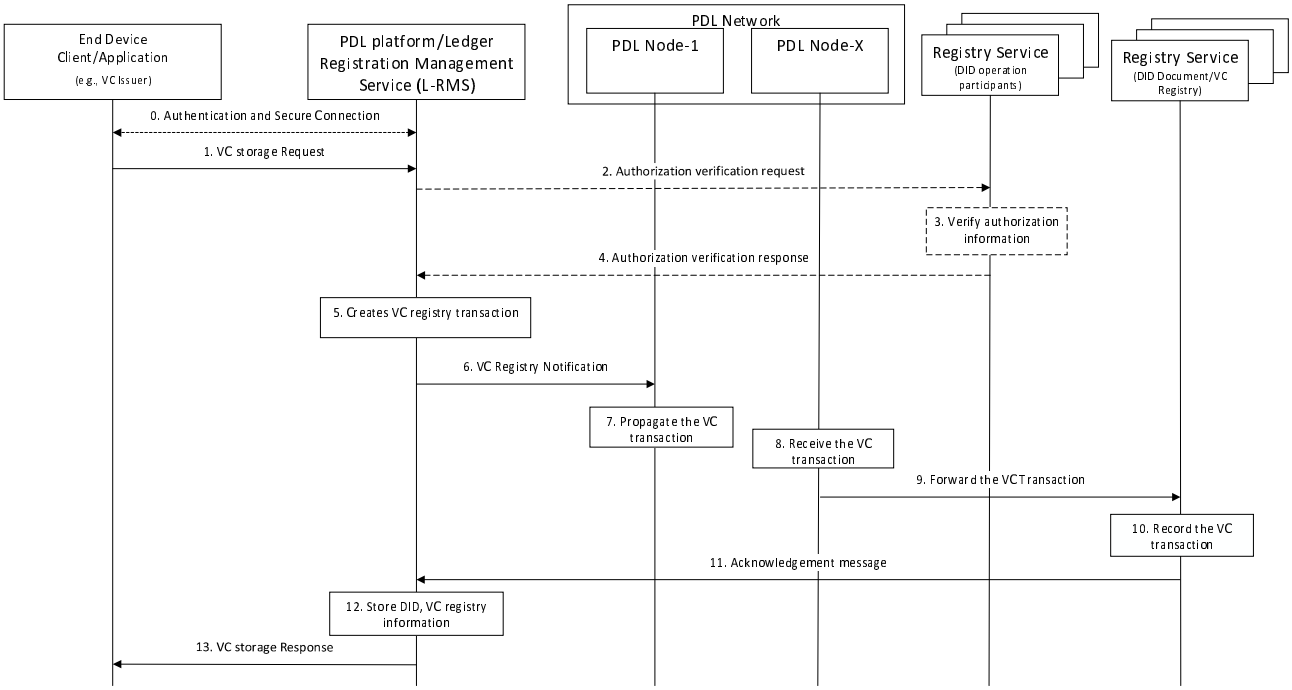


**Figure 5.3.2.4.1-1: VC data management procedure**

#### 5.3.2.4.2 DTMF Operations

##### 5.3.2.4.2.1 DTMF - CREATE Operation

The VC Issuer on generation of VC data, the VC Issuer or the DID holder/controller (after the VC being issued) if wants to use DTMF for storage management aspects specific to CREATE operation, uses the following procedure for the message flow shown in Figure 5.3.2.4.1-1. The create operation helps to store a newly generated and issued VC along with the respective DID:

NOTE 1: The application device client associated to a VC Issuer can be a trust service provider or a DID holder, where DID holder is used to refer both the DID holder/Controller.

0) Precondition: If a secure connection exists, the VC Issuer performs step 1. Else the VC Issuer and the L-RMS performs mutual authentication and sets up a secure connection before step 1, where the authentication can be based on Security Platform Services defined in ETSI GS PDL 012 [1].

1) The end device client/application of the VC Issuer send to the L-RMS, a VC storage request with Registration ID (of the VC Issuer), service type information (i.e. indicates DID service), access role (i.e. indicates VC Issuer), authorization information (i.e. a code or token received as authorization information during a successful registration), DID (of the DID holder), the VC(s), request type (set as Create).

NOTE 2: In case the VC issuer is a trust service provider, the registration ID will be specific to the trust service provider who issues the VC for a DID holder and the DID will indicate the DID holder to which the VC was issued. In case the VC issuer is a DID holder (e.g. self assertation), the registration ID will be specific to the DID holder who issues the VC for itself related to its own DID.

2-4) The L-RMS performs authorization verification specific to the registration ID to verify the access role and authorization information using the DID Operational participant registry service similarly as described in steps 2 to 4 of clause 5.3.2.2. If the authorization verification is successful further step 6 is performed.

Alternatively, for failure case steps 5 to 12 are skipped and step 13 is executed related to the failure case.

5) The L-RMS generates a VC registry notification message with the target Registry service information (i.e. registry service name, and ID (e.g. address) specific to the VC registry or DID Document registry if VC registry managed as part of DID document registry), request type (set as Create), L-RMS ID, Service type information (DID service), Registration ID, authorized access role, DID, and VC(s) (received in step 1). Further the message is transformed into a VC transaction to store the DID and VC(s) to the corresponding registry.

NOTE 3: The message to transaction conversion is up to the PDL platform service provider implementation.

NOTE 4: Based on implementation a DID document registry can also be used to storage a VC or a different VC registry can be managed to store and handle VCs for the DID(s). In case separate registries are maintained for the DID documents and VCs corresponding to a DID, then a smart contract can be implemented to keep track of the DID related DID documents and VC records in different registries.

6) The L-RMS sends to the configured PDL node-1, the VC transaction (which includes the VC registry notification message).

NOTE 5: Based on different implementation, the L-RMS if it is capable to act as a PDL node, it can direct propagate the DID document transaction to the PDL network. In that case, step 7 is skipped.

7) PDL Node-1 propagates the received VC transaction through the target PDL network.

NOTE 6: A group of PDL nodes say PDL 1, 2, and so on e.g. X (that participates in running the protocol software of a PDL network, it includes components such as computing resources e.g. bare metal or virtual machine on which the ledger resides, which also can include wallet, and application) to facilitate consensus, distributed storage, and management of ledger data.

8) PDL Node-X receives the VC transaction from the target PDL network as the result of transaction propagation.

Once VC transaction is validated, it is successfully stored to the ledger (e.g. after PDL consensus process).

9) The PDL Node-X forwards the VC transaction to the VC registry based on the target Registry service information.

10) The VC registry service transforms the transaction into message to recover the message (i.e. VC registry notification message). The VC registry service stores the VC related information (i.e. DID, VCs and other data) received as part of the VC Registry notification message based on local policies, e.g. in a local storage/off-chain.

11) The VC registry service sends to the L-RMS, an acknowledgement message which includes the L-RMS ID (received in step 9 related to the VC registry notification), Registration ID, VC Registry address, and Success.

12) The L-RMS stores the VC Registry address along with DID and other received information locally or in off-chain.

13) The L-RMS sends to the end device client/application, a VC storage response message with DID and Success.

    Alternatively, for the failure case described in step 4, the L-RMS sends to the end device client/application, a VC storage response message with failure indication and a suitable cause value (i.e. such as violation code/authorization failure/authentication failure, etc.).

NOTE 7: Based on different implementation, in Figure 5.3.2.4.1-1, alternatively the L-RMS can be a PDL node by itself and the L-RMS can propagate the PDL transaction (related to the VC data storage management notification) to the PDL network by itself for steps 6 to 8.

#### 5.3.2.4.2.2 DTMF - UPDATE Operation

The update operation helps to update the VC related information over the DTMF as shown in Figure 5.3.2.4.1-1. The procedure described in clause 5.3.2.4.2.1 can be reused for DID specific VC(s) update operations with the following adaptations:

1) In step 1, the end device client/application send to L-RMS, a VC **storage** request with Registration ID, service type information (i.e. it indicates a DID service), access role (i.e. indicates VC Issuer as authorized), authorization information (i.e. a code or token received as authorization information during a successful registration), the ID and updated VC(s), request type (set as Update).

2) In steps 5 to 7, VC notification message (as well as the related transaction) includes the **request type** 'set as Update' (instead of create indication) in addition to the other aspects.

#### 5.3.2.4.2.3 DTMF- DELETE Operation

The delete operation (i.e. revocation) helps to delete/revoke the VC related information over the DTMF as shown in Figure 5.3.2.4.1-1. The procedure described in clause 5.3.2.4.2.1 can be reused for DID specific VC(s) update operations with the following adaptations:

1) In step 1, the end device client/application can send to L-RMS, a VC storage request with Registration ID, service type information (i.e. it indicates a DID service), access role (i.e. indicates VC Issuer as authorized), and authorization information (i.e. a code or token received as authorization information during a successful registration) and request type (set as Delete).

2) In steps 5 to 7, VC registry notification message (as well as the related transaction) includes the **request type** 'set as Delete' (instead of create indication) in addition to the other aspects.

NOTE 1: The smart contracts can be used by the registry services described in this clause to keep track of lifetime expirations and linking of all DID related entries etc. and it is up to the PDL network service operator implementation.

NOTE 2: The L-RMS based on local policy, if identifies that the registered participant has violated any allowed operations (as described in clause 5.3.1.3), it can directly perform step 5 (with DTMF- DELETE Operation specific adaptations) until step 13.

## 5.3.3 Decentralized Identifier Verification Management

### 5.3.3.1 Introduction

This clause describes how the PDL platform based DTMF described in clause 5.2 is used to perform two main aspects such as:

i) DID based DID holder authentication; and

ii) authorization.

Decentralized Identifier verification service described in clause 5.2.2.6 is a composite service [1] which consumes various services offered by DTMF such as DID resolver service, DID document registry service, DID operation(al) participant registry service and VC Data Registry service to fetch DID associated data (such as DID, DID documents and VCs) to enable DID verification by the authorized verifiers (e.g. a service provider). DID Verification process allows DTMF to expose the DID documents and selective data (i.e. based on the associated VCs) to the appropriate verifier for:

i) authentication of the subject identified by the DID; and

ii) authorization check specific to the subject to provide the allowed service(s).

The detailed DID verification process is described further in clause 5.3.3.2.

### 5.3.3.2 DID verification process

The DID verification process is handled using DID verification management service, DID resolver service, DID Operational participants Registry service, DID Document Registry service and VC Data Registry service described in clause 5.2.2 using the steps shown in Figure 5.3.3.2-1.

NOTE 1: Based on the type of DTMF implementation, a DID verification management service and DID resolver service can be co-located or standalone.

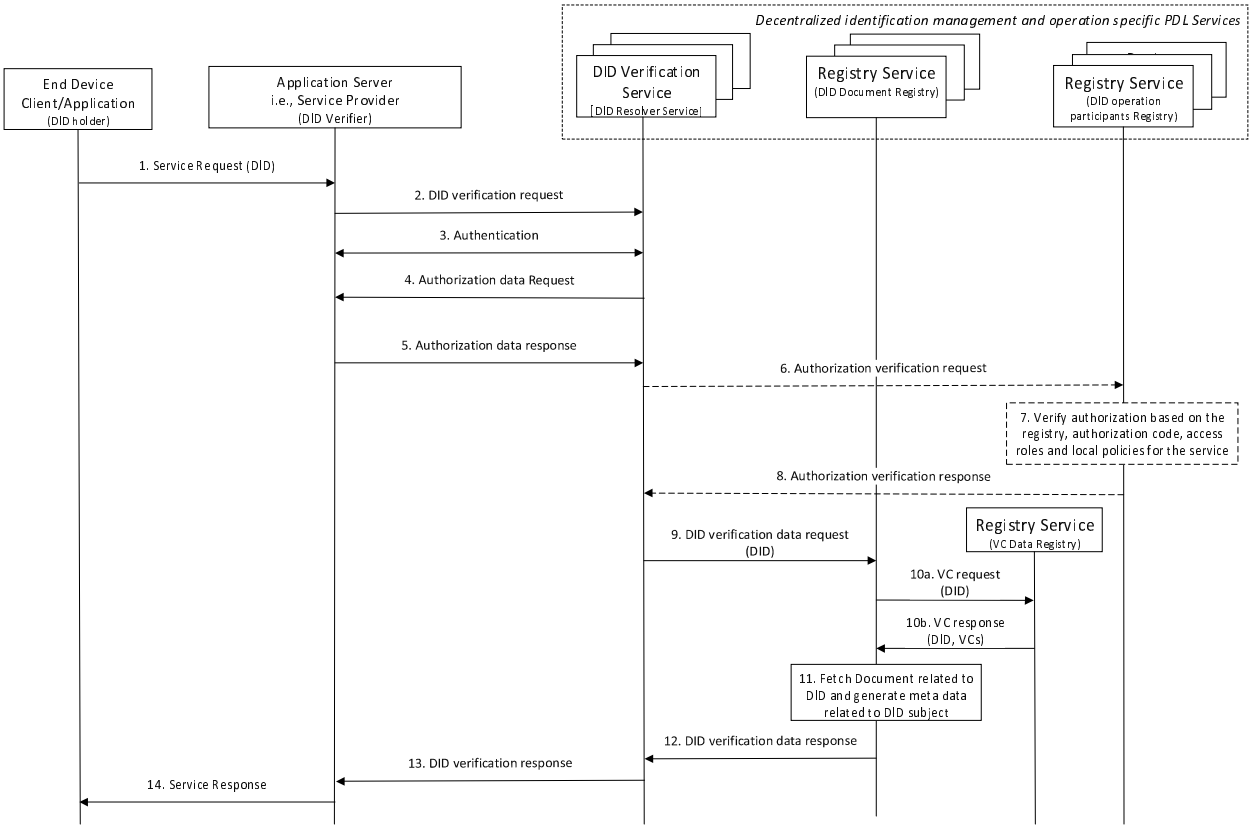


**Figure 5.3.3.2-1: DID verification for DID holder authentication and authorization**

Precondition: The DID holder and the DID Verifier are registered to the DTMF based on clause 5.3.1 to perform various PDL services as required:

1) The DID holder sends to the DID Verifier a service request (or any access request) with DID for a service provision.

NOTE: 2 The service request message that is being sent between the DID holder (i.e. end-device client/application) and the DID verifier (application server/service provider) can be over any interface which is outside the scope. Steps 1 and 14 happen external to the PDL framework, and it is outside the scope of the present document.

2) The DID verifier determine to use the DID Verification Service offered by the DTMF based on the DID (e.g. using realm or information in the DID).

   The DID verifier sends to the DID verification service a DID verification request message with the DID verifier's source ID, DID, target DID service type (i.e. the type of service for which the DID is being associated to the DID holder and being verified by the DID verifier).

3) If there is no secure connection exists between the DID Verifier and the DID verification service, both can perform mutual authentication (using Security Platform Services in ETSI GS PDL 012 [1]) and sets up a secure connection.

4) The DID Verification service sends to the DID Verifier, an Authorization data request message with source identity.

5) The DID Verifier responds to the DID Verification service, it sends an Authorization data response message which includes its Registration ID and its corresponding authorization information (received during role-based registration described in clause 5.2.2.1).

6) The DID Verification service sends to the DID Operation(al) participants Registry service, an Authorization verification request message, which includes the registration ID (of the DID verifier), authorization information (i.e. an authorization code or token as received), access role "set as DID verifier", and for verifier's additionally the service type information (received in step 2).

7) The DID Operation(al) participant registry service verifies the authorization information and access role related to the Registration ID (e.g. by querying the respective ledger for a related transaction records or by checking an offline/local storage) to check if the access role, authorization information and registration ID matches with the records related to the registered participant. For the DID verifiers, additionally the target DID service type information is also checked to see if it is allowed based on the records.

8) If the verification of the registration ID, access role, target DID service type, and authorization information are successful, then the Registry service sends to the DID Verification Service, an authorization verification response message with the registration ID (of the Verifier) and result as "successful".

If the verification of the registration ID, access role, target DID service type, and authorization information do not match with the records, then the Registry service sends to the DID Verification Service, an authorization verification response message, with the registration ID, and result as "failure". Further directly step 13 is performed related to the failure case.

9) The DID Verification services invokes the DID resolver service (i.e. co-located with the DID verification service) and fetches the DID related DID Document Registry address. The DID verification service sends to the DID Document Registry service, a DID verification data request, which includes a DID.

10) a - b:

   a) The DID Document registry service checks if the DID document is available (e.g. in a ledger/chain) for the DID. Further if the DID document is available, the DID document service based on local configuration also finds the VC registry service address and sends to the VC registry service, a VC request message, with the DID.

   b) The VC Registry service fetches the VCs associated to the DID (e.g. from the respective chain/ledger) and sends to the DID document registry service, a VC response message with the DID and the associated VC(s).

11) The DID Document Registry service fetches the DID document(s) related to the DID and generates the VC metadata from the VCs to enable the DID Verifier to authenticate and authorize the DID as required for the service provision.

12) The DID Document Registry service then sends to the DID Verification service, a DID verification data response, with DID, DID documents, and the VC metadata (based on the VCs i.e. claims asserted related to the DID holder specific to the service).

13) The DID Verification service sends to the DID Verifier, a DID Verification response message, which includes result (with successful indication), DID document, and VC metadata (related to VCs).

For the failure case operations, the DID Verification service sends to the DID verifier, a DID verification response message with result set as "failure indication", and cause information (i.e. such as violation code/authorization failure/authentication failure respectively).

14) The DID Verifier can use the DID documents to verify (i.e. to integrity check and authenticate) the DID, and authenticate the DID holder (e.g. related subject). Further the DID verifier can also use the metadata (derived from the VCs) associated to the DID subject to verify the authorization provided for the DID subject, specific to the requested service provision. If the verification of the DID, authentication of the DID subject and the VCs meta data meets the service requirement criteria then the DID verifier sends to the end-device (i.e. DID holder), a Service response message, which can include a successful result. Following which the DID holder will be provided with the requested service. A key associated from the DID document can also be used to set up an initial secure communication between the DID holder and the DID Verifier.

NOTE 3: The metadata based on the VCs can enable to authenticate the subject based on the service specific criteria which are asserted by the claims of the VCs linked to the documents such as passport, driving license, any government issued ID card, college/degree certificate, etc.

EXAMPLE: The VCs may assert claims to prove the service requirements e.g. a service requirement may be the DID holder should be of age above 15 to consume a service (or) the DID holder should belong to a location to consume a service, the DID holder should belong to a country or university or company to consume a service (or) the DID holder should hold a valid driving license to consume a service, etc.

For the failure case operation, the DID Verifier can deny the service request by sending to the end-device (i.e. DID holder), a Service response message with the result as failure and the cause information.

## 5.4 ETSI-ISG-PDL Platform Services

### 5.4.1 Introduction

This clause presents the PDL Platform Services along with the necessary aspects required to realize the PDL based DTMF, and the associated decentralized identification and trust management operations described in clauses 5.2 and 5.3.

### 5.4.2 Ledger Role-based Registration management Services

#### 5.4.2.1 Ledger Role-based Registration management Service - Participant's registration process

[L-RMS-R1] The ETSI-ISG-PDL DTMF Ledger role-based registration management service SHALL support registration of participants to a specific requested and allowed access role.

[L-RMS-R2] The ETSI-ISG-PDL DTMF Ledger role-based registration management service SHALL support registration of participants for DID service.

[L-RMS-O1] The ETSI-ISG-PDL DTMF Ledger role-based registration management service MAY initiate and perform mutual authentication with the entity requesting registration for DID service.

[L-RMS-R3] The ETSI-ISG-PDL DTMF Ledger role-based registration management service SHALL support authentication of the entity requesting registration for DID service.

[L-RMS-R4] The ETSI-ISG-PDL DTMF Ledger role-based registration management service SHALL assign participant registry information with a unique Registration Identifier, Registration Lifetime, allowed access role, and authorization Code for each participant allowed to register for the DID service.

[L-RMS-R5] The ETSI-ISG-PDL DTMF Ledger role-based registration management service SHALL initiate propagation of the participant registry information as a transaction to add the data to the DID Operational Participant Registry service.

[L-RMS-O2] The ETSI-ISG-PDL DTMF Ledger role-based registration management service MAY support propagation of the participant registry information as transaction over the PDL network for validation.

[L-RMS-O3] The ETSI-ISG-PDL DTMF Ledger role-based registration management service MAY support propagation of the participant registry information as transaction over the PDL network (via a specific PDL node) for validation.

[L-RMS-R6] The ETSI-ISG-PDL DTMF Ledger role-based registration management service SHALL inform the entity about the participant registry information if the registration is considered successful.

[L-RMS-O4] The ETSI-ISG-PDL DTMF Ledger role-based registration management service MAY inform the entity about the registration failure based on local policy or if the authentication fails.

#### 5.4.2.2 Ledger Role-based Registration management Service - Participant's de-registration process

[L-RMS-R7] The ETSI-ISG-PDL DTMF Ledger role-based registration management service SHALL support de-registration of participants from a DID service.

[L-RMS-R8] The ETSI-ISG-PDL DTMF Ledger role-based registration management service MAY determine to revoke the registration of a participant based on local policy.

[L-RMS-R9] The ETSI-ISG-PDL DTMF Ledger role-based registration management service SHALL determine to revoke the registration of a participant if it receives from the DID operation participants registry service, a revocation request related to a registration ID.

[L-RMS-R10] The ETSI-ISG-PDL DTMF Ledger role-based registration management service if determines to revoke a registration, it SHALL send to the participant the de-registration notification with Service type information (DID service), Registration ID, and Cause value.

[L-RMS-R11] The ETSI-ISG-PDL DTMF Ledger role-based registration management service May receive from the participant a de-registration response with successful registration revocation acknowledgement.

[L-RMS-R12] The ETSI-ISG-PDL DTMF Ledger role-based registration management service SHALL create registration revocation acknowledgment notification with the L-RMS ID, Service type information (DID service), Registration ID, and Revoked Indication.

[L-RMS-R13] The ETSI-ISG-PDL DTMF Ledger role-based registration management service SHALL initiate propagation of the registration revocation acknowledgment notification as a transaction to add the data to the DID Operational Participant Registry service.

[L-RMS-O5] The ETSI-ISG-PDL DTMF Ledger role-based registration management service MAY support propagation of the registration revocation acknowledgment notification as transaction over the PDL network for validation.

[L-RMS-O6] The ETSI-ISG-PDL DTMF Ledger role-based registration management service MAY support propagation of the registration revocation acknowledgment notification as transaction over the PDL network (via a specific PDL node) for validation.

#### 5.4.2.3 Ledger Role-based Registration management Service - DID and DID Document Storage process

[L-RMS-R14] The ETSI-ISG-PDL DTMF Ledger role-based registration management service SHALL support data storage management related to create, update, and delete operations for DID and DID documents.

[L-RMS-R15] The ETSI-ISG-PDL DTMF Ledger role-based registration management service SHALL initiate authorization verification of the registered participant to allow DID and DID documents storage management.

[L-RMS-R16] The ETSI-ISG-PDL DTMF Ledger role-based registration management service on a successful authorization verification of registered participant SHALL create DID document registry notification with the target Registry service information, request type, L-RMS ID, Service type information (DID service), Registration ID, DID and DID documents.

[L-RMS-R17] The ETSI-ISG-PDL DTMF Ledger role-based registration management service on a successful authorization verification of registered participant SHALL determine to set the request type as create, update, or delete based on the request type received from the entity in the data document storage request.

[L-RMS-O7] The ETSI-ISG-PDL DTMF Ledger role-based registration management service MAY include additional information such as authorized access role and Lifetime in the DID document registry notification.

[L-RMS-O8] The ETSI-ISG-PDL DTMF Ledger role-based registration management service on a successful authorization verification of registered participant MAY set in DID document registry notification the authorized access role as DID holder or DID controller based on the access role received from the entity in the data document storage request.

[L-RMS-R18] The ETSI-ISG-PDL DTMF Ledger role-based registration management service SHALL initiate propagation of the DID document registry notification as a transaction to add the data to the DID Document Registry service.

[L-RMS-O9] The ETSI-ISG-PDL DTMF Ledger role-based registration management service MAY support propagation of the DID document registry notification as transaction over the PDL network for validation.

[L-RMS-O10] The ETSI-ISG-PDL DTMF Ledger role-based registration management service MAY support propagation of the DID document registry notification transaction over the PDL network (via a specific PDL node) for validation.

[L-RMS-R19] The ETSI-ISG-PDL DTMF Ledger role-based registration management service SHALL inform the entity with the DID document registry address and Success indication in the DID document storage response if the DID document storage is considered successful.

[L-RMS-O11] The ETSI-ISG-PDL DTMF Ledger role-based registration management service based on local policy if authentication or authorization fails MAY inform the entity with failure indication and a suitable cause value in DID document storage response.

[L-RMS-R20] The ETSI-ISG-PDL DTMF Ledger role-based registration management service SHALL generate an initial DID document for a DID Holder when the DID Holder requests.

[L-RMS-R21] The ETSI-ISG-PDL DTMF Ledger role-based registration management service SHALL sign the initial DID document using its public key and SHALL send the signed DID document to the DID Holder.

[L-RMS-R22] The signed initial DID document that the ETSI-ISG-PDL DTMF Ledger role-based registration management service sends to the DID Holder SHALL include DID of the DID Holder, DID-Holder-Sign-Auth-Scheme, L-RMS-ID, L-RMS-Sign-Auth-Scheme, and L-RMS-DID-Doc-Sign.

[L-RMS-O12] The signed initial DID document that the ETSI-ISG-PDL DTMF Ledger role-based registration management service sends to the DID Holder MAY include DID-Holder-Public-Key and L-RMS-Public-Key.

[L-RMS-R23] The ETSI-ISG-PDL DTMF Ledger role-based registration management service SHALL receive and verify the initial DID document signed by the DID Holder and from the DID Holder using the DID Holder's public key.

[L-RMS-R24] The ETSI-ISG-PDL DTMF Ledger role-based registration management service SHALL generate a final DID document which contain two signatures: one by L-RMS and one by DID Holder.

[L-RMS-R25] The final DID document SHALL contain the initial DID Document and the verified DID Holder's signature on the initial DID document (DID-Holder-Init-DID-Doc-Sign).

[L-RMS-R26] The ETSI-ISG-PDL DTMF Ledger role-based registration management service SHALL publish the final DID document to a PDL network, which may be indicated by DID Holder or chosen by DID Document Registration Service.

#### 5.4.2.4 Ledger Role-based Registration management Service - VC(s) Storage process

[L-RMS-R27] The ETSI-ISG-PDL DTMF Ledger role-based registration management service SHALL support data storage management related to create, update, and delete operations for VC Data.

[L-RMS-R28] The ETSI-ISG-PDL DTMF Ledger role-based registration management service SHALL initiate authorization verification of the registered participant to allow data storage management for VC Data.

[L-RMS-R29] The ETSI-ISG-PDL DTMF Ledger role-based registration management service on a successful authorization verification of registered participant SHALL create VC registry notification with the target Registry service information, request type, L-RMS ID, Service type information (DID service), Registration ID, VCs.

[L-RMS-R30] The ETSI-ISG-PDL DTMF Ledger role-based registration management service on a successful authorization verification of registered participant SHALL determine to set the request type as create, update, or delete based on the request type received from the entity in the VC storage request.

[L-RMS-O13] The ETSI-ISG-PDL DTMF Ledger role-based registration management service MAY include additional information such as authorized access role in VC registry notification.

[L-RMS-O14] The ETSI-ISG-PDL DTMF Ledger role-based registration management service on a successful authorization verification of registered participant MAY set in VC registry notification the authorized access role as VC Issuer, DID holder, or DID controller based on the access role received from the entity in the VC storage request.

[L-RMS-R31] The ETSI-ISG-PDL DTMF Ledger role-based registration management service SHALL initiate propagation of the VC registry notification as a transaction to add the data to the VC Registry service.

[L-RMS-O15] The ETSI-ISG-PDL DTMF Ledger role-based registration management service MAY support propagation of the VC registry notification as transaction over the PDL network for validation.

[L-RMS-O16] The ETSI-ISG-PDL DTMF Ledger role-based registration management service MAY support propagation of the VC registry notification transaction over the PDL network (via a specific PDL node) for validation.

[L-RMS-R32] The ETSI-ISG-PDL DTMF Ledger role-based registration management service SHALL inform the entity with Success indication in the VC storage response if the VC storage is considered successful.

[L-RMS-O17] The ETSI-ISG-PDL DTMF Ledger role-based registration management service based on local policy if authentication or authorization fails MAY inform the entity with failure indication and a suitable cause value in the VC storage response.

### 5.4.3 DID Operational participants Registry service

#### 5.4.3.1 DID Operational participants Registry service - Participant's de-registration process

[DPRS-R1] The ETSI-ISG-PDL DTMF DID Operational participants Registry service SHALL determine to revoke the participant's registration if it identifies that any participant's registration has expired based on the registration lifetime.

[DPRS-O1] The ETSI-ISG-PDL DTMF DID Operational participants Registry service MAY determine to revoke the participant's registration if a registration expiry is identified and notified by a smart contract or governance to assist the registry service.

[DPRS-R2] The ETSI-ISG-PDL DTMF DID Operational participants Registry service if determines to revoke the participant's registration, it SHALL send to the L-RMS, a revocation request with the registration ID.

[DPRS-R3] The ETSI-ISG-PDL DTMF DID Operational participants Registry service SHALL maintain the participant registry information that is received as transaction notification.

#### 5.4.3.2 DID Operational participants Registry service - Authorization verification process (during DID, DID Document or VC Storage management)

[DPRS-R4] The ETSI-ISG-PDL DTMF DID Operational participants Registry service SHALL verify the authorization information and access role specific to a registration ID on request from the L-RMS.

[DPRS-R5] The ETSI-ISG-PDL DTMF DID Operational participants Registry service on a successful authorization verification SHALL inform the L-RMS about the successful result for the registration ID.

[DPRS-R6] The ETSI-ISG-PDL DTMF DID Operational participants Registry service SHALL inform the L-RMS about the failure result for the registration ID if the access role and authorization information do not match with the records.

#### 5.4.3.3 DID Operational participants Registry service - Authorization verification process (during DID Verification)

[DPRS-R7] The ETSI-ISG-PDL DTMF DID Operational participants Registry service SHALL verify the authorization information, access role specific to a registration ID and service type on request from the DID Verification Service.

[DPRS-R8] The ETSI-ISG-PDL DTMF DID Operational participants Registry service on a successful authorization verification SHALL inform the DID Verification Service about the successful result for the registration ID.

[DPRS-R9] The ETSI-ISG-PDL DTMF DID Operational participants Registry service SHALL inform the DID Verification Service about the failure result for the registration ID if the service type, access role and authorization information do not match with the records.

### 5.4.4 DID Document Registry service

#### 5.4.4.1 DID Document Registry service - DID and DID Document Storage process

[DDRS-R1] The ETSI-ISG-PDL DTMF DID Document Registry service SHALL maintain the DID document information that is received as DID document transaction notification.

[DDRS-R2] The ETSI-ISG-PDL DTMF DID Document Registry service SHALL send DID resolver notification with DID, request type and DID document registry address to the DID resolver service.

[DDRS-O1] The ETSI-ISG-PDL DTMF DID Document Registry service MAY receive DID resolver acknowledgement with DID and success indication.

[DDRS-R3] The ETSI-ISG-PDL DTMF DID Document Registry service on a successful DID document storage SHALL send to the L-RMS an acknowledgement message with Registration ID, DID Document Registry address, Success, and DID resolver registry service ID.

#### 5.4.4.2 DID Document Registry service - DID Verification process

[DDRS-R4] The ETSI-ISG-PDL DTMF DID Document Registry service SHALL fetch the DID specific DID documents locally available, and fetches DID specific VC(s) using the VC Registry service based on the VC registry service address (locally configured).

[DDRS-R5] The ETSI-ISG-PDL DTMF DID Document Registry service SHALL generate the metadata from the VC(s) associated to the DID.

[DDRS-R6] The ETSI-ISG-PDL DTMF DID Document Registry service SHALL provide the DID, DID documents and the VC(s) if it receives the DID verification data request from the DID Verification service.

[DDRS-R7] The ETSI-ISG-PDL DTMF DID Document Registry service SHALL authenticate and authorize DID document publishing request, which L-RMS sends for DID Holders.

[DDRS-O2] The ETSI-ISG-PDL DTMF DID Document Registry service MAY choose a PDL network (PDL-NWK-ID) for a DID holder, where the DID Holder's DID document will be published to.

[DDRS-R8] The ETSI-ISG-PDL DTMF DID Document Registry service SHALL send a DID document publishing authorization response to L-RMS, which may contain DID-Holder-Sign-Auth-Scheme, PDL-NWK-ID, and DID-Holder-Public-Key.

## 5.4.5 DID Resolver service

### 5.4.5.1 DID Resolver service - DID registry

[DRS-R1] The ETSI-ISG-PDL DTMF DID Resolver Registry service SHALL receive, and store DID and DID document registry address if received from DID document registry.

[DRS-O1] The ETSI-ISG-PDL DTMF DID Resolver Registry service on a successful DID document registry address storage specific to a DID MAY send DID resolver acknowledgement with the DID and success indication to the DID document registry.

[DRS-R2] The ETSI-ISG-PDL DTMF DID Resolver Registry service SHALL fetch DID documents using DID document registry address to enable DID verification.

### 5.4.5.2 DID Resolver service - DID Verification process

[DRS-R3] The ETSI-ISG-PDL DTMF DID Resolver Registry service SHALL provide DID associated document registry address to the DID verification service when requested.

## 5.4.6 VC Data Registry service

### 5.4.6.1 VC Data Registry service - VC Data Storage process

[VDRS-R1] The ETSI-ISG-PDL DTMF VC Data Registry Service SHALL maintain the VC data that is received as VC data transaction notification.

[VDRS-R2] The ETSI-ISG-PDL DTMF VC Data Registry Service on a successful VC data storage SHALL send to the L-RMS an acknowledgement message with Registration ID, Success, and VC data registry service ID.

### 5.4.6.2 VC Data Registry service - DID Verification process

[VDRS-R3] The ETSI-ISG-PDL DTMF VC Data Registry Service SHALL on request fetches the VC(s) associated to a DID and provide the VC(s) to the DID document registry service in response.

## 5.4.7 DID Verification Management service

[DVMS-O1] The ETSI-ISG-PDL DTMF DID Verification Management Service MAY initiate and perform mutual authentication with the DID Verifier following the reception of DID verification request.

[DVMS-O2] The ETSI-ISG-PDL DTMF DID Verification Management Service MAY implement DID resolver functionality as a co-located service.

[DVMS-R1] The ETSI-ISG-PDL DTMF DID Verification Management Service SHALL perform authentication of the DID Verifier following the reception of a DID verification request with DID verifier's Source ID, DID (to be verified), target DID service type.

[DVMS-R2] The ETSI-ISG-PDL DTMF DID Verification Management Service SHALL request the Authorization data from the DID Verifier respective identified with the Source ID.

[DVMS-R3] The ETSI-ISG-PDL DTMF DID Verification Management Service SHALL receive the Authorization data from the DID Verifier which includes Registration ID, and authorization information.

[DVMS-R4] The ETSI-ISG-PDL DTMF DID Verification Management Service SHALL initiate authorization verification of the registered participant by using DID Operational participants Registry service to provide the DID Verification Service.

[DVMS-R5] The ETSI-ISG-PDL DTMF DID Verification Management Service SHALL perform DID verification data retrieval from the DID Document Registry (using a standalone or co-located DID resolver service) following the successful authorization verification of the DID verifier.

[DVMS-R6] The ETSI-ISG-PDL DTMF DID Verification Management Service SHALL provide the DID verifier with the necessary data such as successful result, DID documents, and metadata if the DID verification data retrieval is successful.

[DVMS-R7] The ETSI-ISG-PDL DTMF DID Verification Management Service SHALL provide the DID verifier with the failure result, and cause information if any of the authentication, authorization verification or DID verification data retrieval fails.

## 5.5 Summary

The ETSI-ISG-PDL Decentralized Identification and Trust Management Framework defined in the present document provides the architectural enablers and specific list of PDL platform services:

- Ledger Role-based registration management service;
- DID Operational participants Registry service;
- DID Resolver service;
- DID Document Registry service;
- VC Data Registry service; and
- DID Verification management service.

These PDL platform services specified in the present document facilitates the use of end-user/device generated DIDs to be used for authentication and service provision (by the service provider) in such a way that the DID and the associated data exposure required for the service provision will be fully under the control of the DID holder. Any use case (e.g. Telecom service, know your customers service, any digital services, etc.) which requires DID based authentication and trust management shall use the ETSI-ISG-PDL DTMF and the associated the PDL services specified in the present document with sufficient flexibility in the implementations (as needed for the use case).

# Annex A (informative): Change History

| Date | Version | Information about changes |
|---|---|---|
| June 2023 | V0.0.1 | PDL(23)015_009 |
| November 2023 | V0.0.2 | PDL(23)000_156 |
| November 2023 | V0.0.3 | PDL(23)000_171 |
| December 2023 | V0.0.4 | PDL(23)000_180r1 |
| December 2023 | V0.0.5 | PDL(24)017_005 |
| January 2024 | V0.0.6 | PDL(24)017_006r1 |
| April 2024 | V1.1.1 | First published version |

# History

| Document history | | |
|---|---|---|
| V1.1.1 | April 2024 | Publication |
| | | |
| | | |
| | | |
| | | |