ETSI GS PDL 024 V0.1.7 (2024-03)

Permissioned Distributed Ledgers (PDL);

 architecture enhancements for PDL service provisioning in telecom networks

<

**TECHNICAL SPECIFICATION**

Reference

DGS/PDL-0024\_Arch\_Serv\_prov

Keywords

Architecture, Distributed Ledger

***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:
<http://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](http://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 2023.

All rights reserved.

Contents

Intellectual Property Rights 5

Foreword 5

Modal verbs terminology 5

Executive summary 5

Introduction 5

1 Scope 6

2 References 6

2.1 Normative references 6

2.2 Informative references 6

3 Definition of terms, symbols and abbreviations 6

3.1 Terms 6

3.2 Symbols 6

3.3 Abbreviations 7

4 PDL service provisioning architecture model 7

4.1 General concept 7

4.2 Architecture reference model 7

4.2.1 Functional Entities 7

4.2.2 Single-domain reference architecture 7

4.2.3 Ledger data storage reference architecture 9

4.2.4 Architecture to support PDL service information exposure 9

4.2.5 Architecture to support cross-domain PDL service deployment 10

4.2.6 Service-based interfaces 10

4.2.7 Reference points 10

5. High level features of the system 11

5.1 General 11

5.2 PDL Service Management 11

5.3 PDL Service Connectivity Management 11

5.4 PDL Service Security Aspect 11

5.5 PDL Service Performance Assurance 12

5.6 PDL Service Information Exposure 12

5.7 PDL Service Address Management 12

6. Functional Entity 13

6.1 General 13

6.2 Function description 13

6.2.1 DLE 13

6.2.1.1 General Information 13

6.2.1.2 DLE-Client 13

6.2.1.3 DLE-Peer 13

6.2.2 DLAF 13

6.2.2.1 PDL service management 13

6.2.2.2 PDL service operational control 14

6.2.2.2.1 Operational control on DLE 14

6.2.2.2.2 Support operational control on DLDSM 14

6.2.1.2.3 Support operation control on DLRF 14

6.2.3 DLRF 14

6.2.4 DLDSM 15

6.2.5 DLGF 15

7. Function Service Descriptions 15

7.1 General 15

7.2 DLAF Services 15

7.3 DLE Services 17

7.4 DLRF Services 19

7.5 DLDSM Services 20

7.6 DLGF Services 22

8. Procedures for PDL Service Provisioning System 22

8.1 PDL Service Provisioning Procedures 22

8.1.1 PDL Service Description 22

8.1.2 DLE Instantiation 23

8.1.3 PDL Service Deployment 24

8.1.4 PDL Service Update 25

8.1.5 PDL Service Termination 26

8.1.6 DLE Redaction Capability Provisioning 26

8.2 Information Exposure Procedures 28

8.2.1 DLE Information Exposure 28

8.2.1.1 DLE Direct Exposure 28

8.2.1.2 DLE Indirect Exposure 29

8.2.2 DLRF Information Exposure 29

8.2.3 PDL Service Information Exposure 30

8.2.3.1 PDL service internal exposure 30

8.2.3.2 PDL service external exposure via NEF 31

8.2.4 DLDSM Information Exposure 31

8.3 Mobility Management Procedures 32

8.3.1 General Information 32

8.3.2 PDL Service Network Scale-Up 32

8.3.2.1 A new DLE joining in via DLAF 32

8.3.2.2 A new DLE joining in via a peer DLE 32

8.3.3 PDL Service Network Scale-Down 33

8.3.3.1 Direct DLE leaving a PDL service network 33

8.3.3.2 Indirect DLE Leaving a PDL Service network 33

8.4 PDL Service Address Management Procedures 33

9. Integration scenarios in a telecom network with PDL service provisioning capability 33

9.1 General 33

9.2 PDL service provisioning capability native in telecom network 34

9.3 PDL service provisioning capability as a network slicing in telecom network 34

9.4 Deployment Considerations of PDL Service Functions 35

9.4.1 DLAF Deployment Options 35

9.4.2 DLE Deployment Options 35

9.4.3 DLRF Deployment Options 35

9.4.4 LSDM Deployment Options 35

10. Conclusion 35

Annex A (normative or informative): Title of annex 35

Annex (informative): Bibliography 36

History 37

# 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](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 Technical Specification (TS) has been produced by ETSI Technical Committee {ETSI Technical Committee|ETSI Project|<other>} 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](https://portal.etsi.org/Services/editHelp%21/Howtostart/ETSIDraftingRules.aspx) (Verbal forms for the expression of provisions).

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

# Executive summary

# Introduction

# 1 Scope

This document will specify technical solutions for enabling a telecom network to be capable of provisioning various PDL services over the infrastructure itself. The scope of the WI aims to specify required end-to-end enhancements/modifications on:

1) The telecom network architecture across user entities, (radio) access network, core network and service providers (e.g., by adding new functions or modifying functions);

2) Functionalities of the new functions and/or modified functions; and

3) Interfaces and procedures among the new functions and/or existing functions.

# 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 (05-2022): “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]

# 3 Definition of terms, symbols and abbreviations

## 3.1 Terms

For the purposes of the present document, the [following] terms [given in ... and the following] apply:

## 3.2 Symbols

For the purposes of the present document, the [following] symbols [given in ... and the following] apply:

## 3.3 Abbreviations

For the purposes of the present document, the [following] abbreviations [given in ... and the following] apply:

# 4 PDL service provisioning architecture model

## 4.1 General concept

The architecture model for PDL service provisioning is to design the minimum set of functional entities that are required to accommodate and operate a PDL service from a user in the next generation of telecom networks. A user can be either an end user like a UE or an Over-The-Top (OTT) tenant, or even the operator itself. The general concept to design the architecture model is to take into account the constraints of a PLMN such as geographically segmented network domains, distributed infrastructure elements and heterogeneous resource capacities across the entire network infrastructure. Some key concepts are to:

- Modularize the function design.

- Enable each function and its services to interact with other functions and their services directly or indirectly via a Service Communication Proxy if required. The architecture will reuse all available intermediate functions from the underlying PLMN to route CMP messages.

- Wherever applicable, define procedures (i.e. the set of interactions between functions) as services, so that their re-use is possible.

- Support capability exposure.

## 4.2 Architecture reference model

### 4.2.1 Functional Entities

The PDL service provisioning architecture consists of the following functional entities:

* Distributed Ledger Anchor Function (DLAF)
* Distributed Ledger Repository Function (DLRF)
* Distributed Ledger Enabler (DLE)
* Distributed Ledger Data Storage Management (DDLDSM)
* Distributed Ledger Governance Function (DLGF)

### 4.2.2 Single-domain reference architecture

Figure 1 depicts a single-domain PDL service system architecture, where service-based interfaces are used in the PDL service control and management plane:



Figure 1. Single-domain PDL service architecture model with service-based interface in control plane

Figure 2 depicts the single-domain PDL service system architecture.



Figure 2. Single-domain PDL service architecture model with reference point representation

The architecture model represents a scenario where a DLE-Client accesses a PDL service realized by multiple DLE-Peers organized as a distributed ledger (or Blockchain) network embedded in a telecom network. The PDL service connects to a data network (DN). This PDL service is managed and controlled by a set of functional entities at the upper part. In addition, the functional entities can further interact with other telecom network control functions that are typically for existing 3GPP network services.

NOTE 1: A DLE can be a standalone function that is deployed as an individual physical or virtual entity; or a DLE can be a non-standalone function that is co-located with other network functions in the telecom network infrastructure (as shown with the dash box outside). For example, a DLE can co-exist with a user plane function (UPF).

NOTE 2: DLDSM provides external ledger storage capacity if a DLE has limits in capacity or availability time.

NOTE 3: Another PDL service can locate in DN. The existing PDL service running on DLEs can access to the other blockchain network via Nxuf interface. This interface can link to a UPF or a direct connection to DN. A PDL Service in DN can be the same PDL service among participants in one consortium or a different PDL service for inter-blockchain operations.

NOTE 4: The PDL service architecture part may need to interact with network functions (NFs) / entities in the same telecom network for a PDL service provisioning.

NOTE 5: Nxcf interface represents the interactions between DLAF and any other network functions for operational purposes in the same telecom network. The interactions are done by using the services provided by both DLAF and other NFs over specified SBIs.

### 4.2.3 Ledger data storage reference architecture

Figure 3 depicts the architecture model for external storage of the ledger data from DLE. This provides alternatives to a PDL service to offload the ledger data if there are limits on the local DLE such as short of storage or service time termination and so on.



Figure 3. Architecture for ledger data external storage

NOTE 5: DLDSM only handles user service data instead of the operational data. When the service data of a PDL service is offloaded from a DLE to DLDSM (and to an storage), privacy-preserving and data security policies have to be applied.

### 4.2.4 Architecture to support PDL service information exposure

A vertical user shall be able to know the status of a PDL service that is provisioned in a telecom network. The architecture shall be able to expose the information and data of a PDL service to the end user, the tenant or both. This is related to service level agreement (SLA), QoS control or relevant service intervention from an external party.



Figure 4. Architecture for PDL service information exposure

The information of a PDL service can be shared internally and externally via an SBI manner. Specifically, a PDL service running on the provisioned DLEs

### 4.2.5 Architecture to support cross-domain PDL service deployment



Figure 5. Architecture for cross-domain PDL service provisioning

A PDL service can be deployed in multiple PLMN domains. Different domains can refer different operational domains of one PLMN operator, or different network domains of different PLMN operators where the ownership is completely different.

NOTE: A PDL service provisioning is assumed that it shall not be done in a local-break-out (LBO) mode when a DLE-Client roams in a visiting PLMN. Unlike a PDU session, a PDL service semantically involves ledger data that are stored in the PLMN(s) where it is initially provisioned. If a visiting PLMN does not participate the provisioning of that PDL service, a more efficient way is to connect the roaming DLE Client back to its home PLMN and access the PDL service back there. Extending the PDL service temporally to a visiting PLMN requires much longer decision process, which could trigger a lot of overheads in both PLMNs.

### 4.2.6 Service-based interfaces

The PDL service provisioning system architecture contains the following service-based interfaces (SBI):

**Sdlaf**: SBI of DLAF

**Sdlrf:** SBI of DLRF

**Sdldsm:** SBI of DLDSM

### 4.2.7 Reference points

The PDL service provisioning system architecture contains the following reference points:

**Intf1**: Reference point between the PDL-Client and DLAF

**Intf2**: Reference point between the PDL-Client and DLE

**Intf3**: Reference point between the DLAF and DLE

**Intf4**: Reference point between the DLAF and DLDSM

**Intf5**: Reference point between two DLEs

**Intf6**: Reference point between the DLE and DLDSM

**Nxuf**: Reference point between the DLE and user plane connecting to DN

**Intf8**: Reference point between the DLAF and DLRF

**Nxcf**: A generalized interface between the DLAF and other Telecom Network Functions

NOTE 1: The reference point between the DLAF and other telecom network (control) functions reuse the reference points defined in TS 23.501 for interacting with typical network functions (NFs) in a PLMN.

# 5. High level features of the system

## 5.1 General

Clause 5 specifies the high-level functionality and features of the telecom PDL System for both network and applications.

## 5.2 PDL Service Management

The PDL service architecture models shall support the whole lifecycle management of a PDL service from the time the PDL service is request, during its provisioning, deployment and operation, until its termination. In addition, the PDL service architecture shall also support the management of the smart contract intended to be deployed as an application of the PDL service. Specifically, PDL service management includes the following features:

* Handle and parse the PDL service deployment request
* Identify feasible network resources for PDL service deployment request
* Configure network nodes with blockchain capability to become DLEs (e.g., with software libraries, service policies and so on)
* Manage PDL service network topology (e.g., topological structure, links among DLEs and so on)
* Configure DLE’s profile for a PDL service (e.g., consensus protocol, redaction policy and participating roles)
* Review and publish smart contract of a PDL service (e.g., compatibility, validness and threat analysis of a smart contract)

## 5.3 PDL Service Connectivity Management

The PDL service architecture models shall maintain the connectivity of a DLE to a provisioned PDL service especially when mobility effects, which could be either the mobile DLE-Client or a DLE-Peer running on a mobile node, influence the reliability of the network connection of a deployed PDL service. The functional entities of the PDL service architecture part shall interact with the related NFs responsible for the existing telecom network services to monitoring the connectivity status of a DLE in case any adaptation of the provisioned PDL service. Specifically, PDL service connectivity management includes the following features:

* Establish connection (e.g., under the instruction of DLAF) for both a DLE-Client to a PDL service node (i.e., a DLE) and connections among multiple DLE-Peers
* Monitor connectivity of a DLE-Client for accessing a deployed PDL service (e.g., bandwidth, delay and so on)
* Monitor connectivity of DLE-Peers contributing to a provisioned PDL service (e.g., bandwidth, delay and so on)
* Conduct DLE-Client connectivity update with interacting and coordinating with other functional entities (e.g., existing NFs in the telecom network)
* Conduct DLE-Peer connectivity update with interacting and coordinating with other functional entities (e.g., existing NFs in the telecom network)

## 5.4 PDL Service Security Aspect

The PDL service architecture shall handle the security aspect of any PDL service that is requested, provisioned and operated in the telecom network for both the PDL service provider side and the user side (e.g., DLE-Clients). This includes the security guarantee procedures for accommodating a PDL service of a PDL service provide and full access control of a PDL service user. Specifically, PDL service security aspect includes the following features:

* Generate and distribute cryptographical materials for both a DLE-Client and DLE-Peers of a PDL service (by coordinating with DLGF if necessary)
* Authenticate and authorize a DLE-Client when accessing the PDL service (by coordinating with DLGF if necessary)
* Authenticate and authorize a DLE-Peer which joins in as a new DLE-Peer to contribute a PDL service (by coordinating with DLGF if necessary)
* Enforce confidentiality and integration for PDL service data including user data (e.g., identity and generated ledger data), signaling data between any two entities such as DLE-Client, functional entities (e.g., DLAF) and DLE-Peers
* Configure security policies for both DLE-Client and functional entities (e.g., DLEs，DLRF，DLDSM) with the guidance from governance layer (e.g., DLGF)
* Analyze abnormal/malicious behaviors of a deployed PDL service; if needed, trigger to prescribe corresponding mitigation strategies (by coordinating with other functional entities
* Provide trust execution environment (TEE) for operating smart contract and distributed consensus mechanism

## 5.5 PDL Service Performance Assurance

The PDL service architecture shall guarantee the performance of a provisioned PDL service that is mutually agreed with the owner of the PDL service. The PDL service architecture shall support performance monitoring of a deployed PDL service where the service performance running on all involved DLEs can be monitored and performance metrics can be collected. In addition, with the collected performance metrics, the system shall be able to decide if the deployment configuration of a PDL service has to be updated. With coordinating with other functional entities (e.g., DLGF) PDL service performance assurance includes the following features:

* Collect key performance indicators (KPIs) of a deploy PDL service
* Analyze collected performance measurements and trigger service assurance adaptation process
* Execute service update subject to performance constraints (e.g., DLE addition or removal, DLE configuration update, DLE migration and service scheduling)

## 5.6 PDL Service Information Exposure

The PDL service architecture shall support to expose information related to a PDL service requested by internal and/or external consumers. Performance status information can be required for both internal and external users to monitor the status of a deployed PDL service. For example, an internal user such as another functional entity can subscribe the performance status from DLEs; with the collected information, a functional entity can determine if any action is needed to adjust the running PDL service, or characterize the profile of the PDL service for trend analysis and so on. For external user, such as the owner of the PDL service, the actual service provider can subscribe the information in order to determine if external intervention is needed. In addition, an exposure mechanism is also required to enable the information subscription and notification between the producer and consumer. PDL service information exposure includes the following features:

* Collect service information of a deployed PDL service in the telecom network
* Provide relevant PDL service event for performance measurement, resource consumption, sustainability metrics and so on
* Provide exposure interfaces for information subscription and notification of a PDL service for both internal and external parties

## 5.7 PDL Service Address Management

DLE-Client needs to have a blockchain address in order to access services provided by DLE (e.g., send a transaction to DLE). A DLE also has a blockchain address, which is used to send transactions to distributed ledgers for control and management purpose.

Those blockchain addresses shall be permissioned as a part of PDL system. DLGF coordinates blockchain address generation and authentication.

* DLGF configures and instructs DLE-Clients and DLEs how to generate their blockchain address.
* DLE-Clients and DLEs present their blockchain addresses to DLGF for authorization, before they can access PDL services.
* DLGF authenticates and authorizes the blockchain address of a DLE-Client (or a DLE). During this process, DLGF may need to request key materials about the DLE-Client from the 3GPP network (e.g., AUSF or SEAF).
* DLGF sends the blockchain address authorization response to the DLE-Client (or the DLE), in which configuration instructions for the DLE-Client may be included if the blockchain address is successfully authorized.
* The DLE-Client (or the DLE) follows the configuration instructions and starts to access PDL services after its blockchain address is authenticated and authorized.

# 6. Functional Entity

## 6.1 General

In this section, the specific functional features are defined for all functional entities in the proposed PDL service provisioning architecture.

## 6.2 Function description

### 6.2.1 DLE

#### 6.2.1.1 General Information

Distributed Ledger Enabler (DLE) is the main element, in which a PDL service is deployed.

#### 6.2.1.2 DLE-Client

In this mode, DLE does not participate any consensus or validation process. It acts as a client interfacing to the end user/device/NF for local transaction composition and submission; in addition, it also interacts with DLAF for control and management plane signalling. An example is that a DLE-Client is installed on a UE as an App where transaction traffics are sent out.

#### 6.2.1.3 DLE-Peer

DLE-Peer: In this mode. DLE may participate consensus or validation process, where the extent depends on the local capability. In this mode, a DLE-Peer can act in the following specific modes:

* Micro Mode: This mode has the ability to accept and verify transactions submitted by the client, compose transactions and package them into micro blocks, and broadcast them to other peers. However, a DLE-Peer in micro mode does not participate consensus process but only synchronize the consensus results;
* Lightweight Mode: In addition to the capability of a DLE-Peer in micro mode, a DLE-Peer in lightweight mode has the ability to validate micro blocks participates consensus process. However, a DLE-Peer in lightweight mode does not necessarily store full ledger data
* Full Mode: This mode contains all capabilities of the lightweight mode. In addition, a DLE-Peer in full mode will keep the ledger data with its local storage

In addition to the functions, all three modes have dynamic topology maintenance function, wherein two ways are supported as below:

* Passive: The topology information, e.g., the addresses of neighbouring DLE-Peers, is fully provided by controlling functional entities
* Autonomic: The topology information, e.g., the addresses of neighbouring DLE-Peers, is autonomically discovered by a DLE-Peer

### 6.2.2 DLAF

The main functions of DLAF are categorized into two aspects below.

#### 6.2.2.1 PDL service management

The PDL service management on DLAF includes:

* Identifying feasible network resources in a telecom network infrastructure.
* Preparing network nodes with DLE initial instantiation, where major tasks are listed as follows:
	+ Activate software components for a PDL service on a network node if the software components are already available on the network node
	+ Deactivate the software components on a network node to terminate a DLE instance, which could be reactivated again if needed
	+ Remove unnecessary software components that are not required for a PDL service. After being removed, the software component will be unavailable on the network node
	+ Instruct a network node to download/update software components (from DLRF) for a PDL service if the necessary software components are unavailable/not up-to-date on the network node
	+ Lock a DLE instance on a network node in order to prevent from unwanted configurations (e.g., spontaneous software updates and malicious access) when the DLE is in use
* Preparing instantiated DLE with specific PDL service capability required for a PDL service, where major/non-exhausted tasks are listed as follows:
	+ Configure one or multiple distributed consensus protocols, related security algorithms and so on
	+ Deploy service policies such as permissions to read/write/query, DLE’s participating roles
	+ Provide smart contract templates for composing decentralized applications (DAPPs)
* Registering all DLE profiles and responding the lookup requests from others

#### 6.2.2.2 PDL service operational control

##### 6.2.2.2.1 Operational control on DLE

PDL service operational control on DLAF includes:

* Create a PDL service network, where the major tasks are listed as follows:
	+ Receive and analyse a PDL service provisioning request (e.g., the number of DLE instances needed, strength of security level with a threat model, performance requirement and reliability level)
	+ Coordinate with other functional entities (e.g., DLGF) for generating and distributing crypographical materials for every DLE-Client and DLE-Peer, which is the end user of the concerned PDL service
	+ Select DLE instances (including DLE-clients and DLE-Peers) from network resource pool subject to the PDL service provisioning request in terms of the PDL capabilities available from the DLE instances, network performance metrics, security and reliability considerations;
	+ Activate selected DLE instances on the selected network resource nodes and establish the topological connectivity among the selected DLE instances
	+ Configure off-DLE ledger data storage policy
* Conduct access control for a DLE-Client when the DLE-Client requests to access a PDL service
* Conduct access control for a new DLE-Peer instance is selected to participate an existing PDL service network
* Monitor operational status of a provisioned PDL service including transaction confirmation speed, the loads on every PDL instances in the PDL service network and potential anomality/attacks
* Provide information exposure service interfaces for a deployed PDL service

##### 6.2.2.2.2 Support operational control on DLDSM

DLAF shall configure DLDSM with data storage policy for a PDL service provisioning request. This includes:

* Access permissions for DLE Peers (e.g., in terms of the participating roles of DLE Peers, time periods and so on)
* Privacy policies of off-DLE Peer storage (e.g., encryption/decryption and anonymization methods)
* Exposure policy for ledger data query (e.g., white/black lists of legitimate requesters)
* Storage capacity allocation policy (e.g., for DLE instances of a PDL service and storage capacity allocations among different PDL services)

##### 6.2.1.2.3 Support operation control on DLRF

DLAF shall configure DLRF for the following (non-exhausted) tasks:

* Initialize and update the software libraries needed for PDL services
* Trigger to validate the correctness of the stored software libraries
* Define the access policy

### 6.2.3 DLRF

Distributed Ledger Repository Function (DLRF) is a repository function providing required software components for the realization of a PDL service when DLAF controls to manage and configure PDL capability (e.g., required software components on DLEs for a PDL service) within the telecom network infrastructure.

The main function of a DLRF is to provide necessary software to a resource node whenever the necessary software capability is missing on the resource node. Specifically, a resource node sometime is not installed with all software that is needed to run a BC network, such as the distributed consensus protocol. Therefore, when a resource node is selected while a certain software is missing, DLAF will retrieve the software and install it on the targeted resource node. After that the resource node is capable of running the PDL service.

In general, DLRF collects software libraries, toolkits and binary codes of popular BC realizations. Specifically, first, DLRF provides software libraries of various distributed consensus protocols (e.g., PBFT, RAFT, PAXOS and so on), from which DLAF can pick one consensus protocol that a PDL service provisioning request specifies. In addition, DLRF provides standardized data structures of ledger organization, transaction (block) header format (such as the Merkel tree or Trie implementation libraries); furthermore, DLRF provides software libraries for hash function (e.g., Chameleon, MD5 and SHA-256), encryption (e.g., RSA, ECC and Lattice) and digital signature algorithms (e.g., DSA and ECDSA).

### 6.2.4 DLDSM

Ledger Data Storage Management (DLDSM) is a broker function to the actual storage capability of a telecom network. It accepts the request from a DLE-Peer node who transfers the local ledger data to another storage location external to the DLE-Peer node. In addition, DLDSM is also responsible for retrieving requested ledger data from another authorized consumer entity or even an external party. Accessing the archived ledger data shall first request to DLAF (interacting with BCGF if necessary) and DLAF will authorize the access permission to DLDSM, or DLAF request the DLDSM to authorize the access permission, or DLAF send the request to DLDSM. Any ledger data handled by DLDSM shall follow the privacy-preserving and data security policies of the whole system.

### 6.2.5 DLGF

DLGF is a functional entity to coordinate and govern all PDL functional entities (i.e., DLAF, DLRF, DLE, DLDSM) as defined in the present document. In principle, DLGF implements Governance Platform Services (GPS) as defined in clause 4.6.3.6 of ETSI GS PDL 0012 [1]. According to [1], GPS is a collection of rules and tools that control the behaviour and function of a PDL Platform. GPS is divided to two functions:

* Implementation Agreements (IAs): A collection of rules and agreements that describe how services are implemented and control the behavior of the PDL platform.
* Governing Entity: An entity that performs governance tasks by defining the rules and IAs, as well as ensuring compliance and resolving conflicts where needed. Governance also defines the methods by which the Governing Entity is established, its composition and the methods by which it defines/accepts rules/IAs and enforces compliance.

DLGF expands GPS with the following additional functionalities:

* Authenticate and authorize if an externa entity (e.g., DLE-Client, a Telecom NF, etc.) is allowed to request and access services provided by PDL function entities as defined in the present document.
* Coordinate interactions among PDL function entities as defined in the present document.
* Coordinate and manage underlying distributed ledger networks such as the management of PDL nodes.
* Coordinate and manage provisioning ledger redaction capabilities to PDL function entities.

# 7. Function Service Descriptions

## 7.1 General

In the context of this specification, a proposed functional entity offers a capability to authorised consumers. The new functions may offer different capabilities and thus, different function services to distinct consumers. Each of the function services offered by the new functions shall be self-contained, reusable and use management schemes independently of other NF services offered by the same Network Function (e.g. for scaling, healing, etc).

## 7.2 DLAF Services

The following services are specified for DLAF:

Table 1. List of DLAF Services

|  |  |  |  |
| --- | --- | --- | --- |
| **Service Name** | **Description** | **Service Operations** | **Consumer Entities** |
| Ndlaf\_PdlFuncEntity | This service manages other functional entities in the PDL service provisioning architecture (e.g., DLE-Peer, DLRF, DLDSM and so on)  | Activate | Any governance function that manages the functional entities for PDL service provisioning |
|  |  | Deactivate |  |
|  |  | download |  |
|  |  | Add |  |
|  |  | Remove |  |
|  |  | unlock |  |
|  |  | Lock |  |
| Ndlaf\_PdlService | This service configures a PDL service with DLE-Peers and manages the whole life circle of a deployed PDL service | Create | A control function that deploys a PDL service |
|  |  | Stop |  |
|  |  | Update |  |
|  |  | Remove |  |
| Ndlaf\_InfoExposure | This service provides interfaces to subscribe / unsubscribe event information | Subscribe | Any NF or functional entity that is interested a particular event of a deployed PDL service |
|  |  | Unsubscribe |  |
|  |  | Notify |  |
|  |  | Transfer |  |

Ndlaf\_PdlFuncEntity collects a set of services provided by DLAF. The main job of this service interface is to let the consumer entity to operate the other functional entities for PDL service provisioning. This service interface at least contains the following concrete operations to the functional entities for PDL service.

* Ndlaf\_PdlFuncEntity\_Activate: This service interface allows a consumer entity to activate a specific functional entity for PDL service provisioning. This functional entity is one of the entities proposed in the architecture models, excedpt DLGF.
* Ndlaf\_PdlFuncEntity\_Deactivate: This service interface allows a consumer entity to deactivate a specific functional entity for PDL service provisioning. This functional entity is one of the entities proposed in the architecture models, except DLGF.
* Ndlaf\_PdlFuncEntity\_Download: This service interface allows a consumer entity to download a specific software for a functional entity for PDL service provisioning. This functional entity is one of the entities proposed in the architecture models, except DLGF.
* Ndlaf\_PdlFuncEntity\_Add: This service interface allows a consumer entity to add a new functional entity instance for PDL service provisioning in the telecom network domain. This functional entity is one of the entities proposed in the architecture models, except DLGF.
* Ndlaf\_PdlFuncEntity\_Remove: This service interface allows a consumer entity to remove an existing new functional entity instance for PDL service provisioning in the telecom network domain. This functional entity is one of the entities proposed in the architecture models, except DLGF.
* Ndlaf\_PdlFuncEntity\_Lock: This service interface allows a consumer entity to lock a functional entity instance for PDL service provisioning in the telecom network domain. This will maintain the functional entity but temporally prohibit it actions, which makes this entity unavailable. This functional entity is one of the entities proposed in the architecture models, except DLGF.
* Ndlaf\_PdlFuncEntity\_Unlock: This service interface allows a consumer entity to unlock a functional entity instance for PDL service provisioning in the telecom network domain. This will bring back a functional entity that was temporally suspended before and make this entity available again. This functional entity is one of the entities proposed in the architecture models, except DLGF.

Ndlaf\_PdlService collects a series of interfaces provided from DLAF where a consumer entity can operate with a PDL service that will deployed within a telecom network. This service interface at least contains the following concrete operations to the functional entities for PDL service.

* Ndlaf\_PdlService\_Create: This interface allows a consumer entity to create a PDL service via DLAF consisting of a set of participating DLE instances. The participating DLE instances are the nodes processing the PDL transactions submitted by the users (with running a specified distributed consensus protocol)
* Ndlaf\_PdlService\_Stop: This interface allows a consumer entity to stop a PDL service via DLAF in the telecom network. This will stop the executions on the participating DLE instances that were originally assigned to load the PDL service.
* Ndlaf\_PdlService\_Update: This interface allows a consumer entity to update the configuration of a PDL service via DLAF in the telecom network. This will trigger updates on one or multiple participating DLE instances. The update can be the behaviours of the participating DLE instances (e.g., updating the distributed consensus protocol, updating the role of the DLE instances and so on).
* Ndlaf\_PdlService\_Remove: This interface allows a consumer entity to remove an existing PDL service that has been deployed on one or multiple DLE instances in the telecom network. This may or may not eventually remove the DLE instances together with the terminated PDL service itself, depending on the specific parameters sent with calling the service.

## 7.3 DLE Services

The following services are specified for DLEF:

Table 2. List of DLE Services

|  |  |  |  |
| --- | --- | --- | --- |
| **Service Name** | **Description** | **Service Operation** | **Consumer Entities** |
| Ndle\_InfoExposure | This service enables to subscribe and notify event information with data of an interested PDL service on a DLE | Subscribe | An NF or functional entity that is interested in one or multiple events on DLE-Peers |
|  |  | Unsubscribe |  |
|  |  | Notify |  |
| Ndle\_DataTransfer | This service provides interfaces to allow a DLE to make interaction with another DLE about ledger data operations. For example, a DLE can lookup a data record on another DLE and/or synchronize a data record with another DLE. | Lookup | DLE |
|  |  | Sync |  |
| Ndle\_PdlConnection | This service provides functions related to DLE connections, including performing node discovery to establish blockchain topology; dynamically establishing connections between DLE nodes and periodically performing connection checks | Discover | DLE |
|  |  | Connect | DLE |
| Check | DLE |
| Ndle\_Capability | This service provides all operations to manage the capability of a DLE and manages the configurations to a DLE for a requested PDL service including configuring the mode of a DLE, issuing certificates and operation status | Activate | DLAF |
| Deactivate | DLAF |
|  |  | Download | DLAF |
|  |  | Remove | DLAF |

Ndle\_InfoExposure collects a series of service interfaces where a consumer entity can use to monitor the behaviours/statuses on a DLE instance. This service interface at least contains the following concrete operations to the functional entities for PDL service.

* Ndle\_InfoExposure\_Subscribe: This interface allows a consumer entity to subscribe an interested event from a DLE instance. This interface aims to expose the information (such as the running statuses, exceptions and so on) generated from a DLE instance to another party, either internal or external the telecom network domain.
* Ndle\_InfoExposure\_Unsubscribe: This interface allows a consumer entity to unsubscribe an interested event from a specified DLE instance that was previously subscribed.

Ndle\_DataTransfer collects a series of service interfaces that allow a consumer entity to do local data operations on a DLE instance. This service interface at least contains the following concrete operations:

* Ndle\_DataTransfer\_Lookup: This interface allows a consumer entity to lookup/read a specific data information from the ledger stored on a DLE instance.
* Ndle\_DataTransfer\_Sync: This interface allows a consumer entity to command a specific DLE instance synchronizing the ledger data with another authorized functional entity.

Ndle\_PdlConnection collects a series of service interfaces that allow a consumer entity to operate the connections where a DLE instance connects to other DLE instances. This will operate the topological structure of the PDL service. It at least includes the following service interfaces:

* Ndle\_PdlConnection\_Discover: This interface allows a consumer entity to command a DLE instance to discover one or multiple neigbhor DLE instances in order to form a PDL service DL network structure.
* Ndle\_PdlConnection\_Connect: This interface allows a consumer entity to command a DLE instance to establish a connection to another DLE instance. This will lead either of the DLE instance to join in the service network that provisions an existing PDL service.
* Ndle\_PdlConnection\_Check: This interface allows a consumer entity to command a DLE instance to check the connectivity status of a link from the DLE instance to another DLE instance. This can be used before, during and/or after a PDL service provisioning when the link status information needs to be known for other decision-making tasks.

Ndle\_Capability collects a series of service interfaces that allow a consumer entity to configure the behaviours of a DLE instance. This at least includes the following service interfaces:

Ndle\_Capability\_Activate: This interface allows a consumer entity to command a DLE instance to activate a certain capability for a specific PDL service on the DLE instance. The capability includes any configuration specific for processing the transactions that are submitted to the DLE instance.

Ndle\_Capability\_Deactivate: This interface allows a consumer entity to command a DLE instance to deactivate a certain capability for a specific PDL service on the DLE instance. The capability includes any configuration specific for processing the transactions that are submitted to the DLE instance.

Ndle\_Capability\_Download: This interface allows a consumer entity to command a DLE instance to download a certain software library/patch/configuration parameters/cryptographic materials for a specific PDL service on the DLE instance.

Ndle\_Capability\_Remove: This interface allows a consumer entity to command a DLE instance to remove a certain software library/patch/configuration parameters/cryptographic materials for a specific PDL service on the DLE instance.

## 7.4 DLRF Services

Table 3. List of DLRF Services

|  |  |  |  |
| --- | --- | --- | --- |
| **Service Name** | **Description** | **Service Operation** | **Consumer Entities** |
| Ndlrf\_Policy | This service provides interfaces to configure the storage policy, access policy (e.g., whether allowing a direct access from a DLE), format policy (e.g., binary file, executable file and/or source code) and so on. | Configure | DLAF |
|  |  | Delete |  |
|  |  | Update |  |
| Ndlrf\_Library | This service allows an authorized consumer function to look up a specific software library. This retrieved software library (in a form of URL or binary data package) will be used to configure a set of resource nodes in the network for PDL service provisioning. This service allows a management entity to update the library for realizing PDL services on DLRF including version update, obsolete library removal and so on. | Lookup | DLAF, DLE |
|  |  | Insert |  |
|  |  | Delete |  |
|  |  | Update |  |
| Ndlrf\_InfoExposure |  | Subscribe |  |
|  |  | Unsubscribe |  |
|  |  | Notify |  |

Ndlrf\_Policy provides a series of service interfaces to a consumer entity for interacting with DLRF for configuring the policy for PDL service provisioning in a telecom network. It at least contains the following service interfaces:

* Ndlrf\_Policy\_Configure: This service interface allows a consumer entity to add a specific policy for a PDL service. This policy can be any policy that can influence the way how the telecom network shall provide a PDL service. For example, the policy can be related about service level agreement (SLA), security, provisioning rules and so on. The main consumer entity will be DLGF.
* Ndlrf\_Policy\_Delete: This service interface allows a consumer entity to remove a policy from DLRF for a PDL service. This policy can be any policy that can influence the way how the telecom network shall provide a PDL service. For example, the policy can be related about service level agreement (SLA), security, provisioning rules and so on. The main consumer entity will be DLGF.
* Ndlrf\_Policy\_Update: This service interface allows a consumer entity to change a policy that exists in DLRF for a PDL service. This policy can be any policy that can influence the way how the telecom network shall provide a PDL service. For example, the policy can be related about service level agreement (SLA), security, provisioning rules and so on. The main consumer entity will be DLGF.

Ndlrf\_Library provides a series of service interfaces that allow a consumer entity to interact with DLRF for the software library configurations.

* Ndlrf\_Library\_Lookup: This interface allows the consumer entity to check if a certain software library already exists in the DLRF. The software library could be any software that is required for running a PDL service on DLE.
* Ndlrf\_Library\_Insert: This interface allows the consumer entity to add a certain software library that does not exist in the DLRF. The software library could be any software that is required for running a PDL service on DLE.
* Ndlrf\_Library\_Delete: This interface allows the consumer entity to delete a certain software library that already exists in DLRF.
* Ndlrf\_Library\_Update: This interface allows the consumer entity to update an existing software library that already exists in DLRF.

## 7.5 DLDSM Services

Table 4. List of DLDSM Services

|  |  |  |  |
| --- | --- | --- | --- |
| **Service Name** | **Description** | **Service Operations** | **Consumer Entities** |
| Ndldsm\_Policy | This service provides interfaces to configure the storage policy, access policy (e.g., whether allowing a direct access from a DLE), security policy (e.g., encryption, signature and integrity) and so on | Add | DLAF |
|  |  | Delete |  |
|  |  | Update |  |
| Ndldsm\_LgData | This service provides all operations for the ledger data of a DLE including save, update and delete | Save | DLAF, DLE |
|  |  | Delete |  |
|  |  | Modify |  |
|  |  | Lock |  |
|  |  | Unlock |  |
|  |  | Lookup |  |
| Ndldsm\_Exposure | This service provides all operations such as subscribe, unsubscribe and notify for the ledger data exposure from DLDSM to an authorized ledger data consumer | Subscribe | DLAF, DLE |
|  |  | Unsubscribe |  |
|  |  | Notify |  |

Ndldsm\_Policy provides a series of services for configuring the policies that will be applied when handling the ledger data stored on DLDSM. These services are mainly used by DLGF. It at least contains the following service interfaces:

* Ndldsm\_Policy\_Add: This service interface allows a consumer entity to add a new policy on DLDSM for ledger data storage. For example, a policy that restricts the access right for a certain consumer, a policy that defines the access attributes of the ledger data and so on.
* Ndldsm\_Policy\_Delete: This service interface allows a consumer entity to delete an existing policy on DLDSM.
* Ndldsm\_Policy\_Update: This service interface allows a consumer entity to update an existing policy on DLSDM. For example, a consumer entity may modify the scope of the access right of specific ledger data.

Ndldsm\_LgData provides a series of services for ledger data operations on DLDSM. These services are mainly used by DLE. It at least contains the following service interfaces:

* Ndldsm\_LgData\_Save: This interface allows an authorized consumer entity to save a piece of ledger data out-of-band from the distributed ledger where the ledger data was generated. The offline stored ledger data should contain verifiable metadata about the data source and ownership.
* Ndldsm\_LgData\_Delete: This interface allows an authorized consumer entity to delete a piece of ledger data out-of-band from the distributed ledger where the data was generated.
* Ndldsm\_LgData\_Modify: This interface allows an authorized consumer entity to modify a piece of ledger data out-of-band from the distributed ledger where the data was generated.
* Ndldsm\_LgData\_Lock: This interface allows an authorized consumer entity to temporally lock a piece of ledger data on DLSDM. Locking the ledger data prohibits data retrieval from a consumer entity but the existence of the ledger data is still visible.
* Ndldsm\_LgData\_Unlock: This interface allows an authorized consumer entity to unlock a piece of ledger data on DLSDM that was previously locked.
* Ndldsm\_LgData\_Lookup: This interface allows an authorized consumer entity look up a piece of ledger data with keywords.

Ndldsm\_Exposure provides a series of services to expose the ledger data on DLSDM to other entities. It at least contains the follow service interfaces:

* Ndldsm\_Exposure\_Subscribe: This interface allows an authorized entity to subscribe events related to the ledger data status. For example, an entity can subscribe an event that a certain type of ledger data is being created on DLDSM, and/or the data is being removed from DLDSM.
* Ndldsm\_Exposure\_Unsubscribe: This interface allows an authorized entity to unsubscribe events related to the ledger data status that was previously subscribed by the entity.
* Ndldsm\_Exposure\_Notify: This interface allows DLDSM notify an functional entity with an event with event data information. The notified functional entity is the consumer who previously subscribes the occurring event.

## 7.6 DLGF Services

Table 5. List of DLGF Services

|  |  |  |  |
| --- | --- | --- | --- |
| **Service Name** | **Description** | **Service Operations** | **Consumer Entities** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# 8. Procedures for PDL Service Provisioning System

## 8.1 PDL Service Provisioning Procedures

### 8.1.1 PDL Service Description

A PDL service is described with the following information and a PDL service provisioning request shall contain a service description information and is provided to a telecom network. The service description information is summarized in the following table.

Table 6. PDL Service Description

|  |  |  |  |
| --- | --- | --- | --- |
| **Classification** | **Attribute Name** | **Example** | **Description** |
| General Property | Player ID | ID1, ID2, ID3, … | The identifiers of all participants forming the PDL service |
| Ledger Topology | Single-chain, multi-chain, or DAG | The topology structure organizing ledger transaction data |
| Consensus Protocol | PoS, PoW, Raft, PBFT | The consensus protocol option(s) that shall be used for this PDL service |
| Transaction per second (TPS) | 100, 500, 1000 | Required throughput of the PDL service |
| Redactable | Yes or No | Whether the PDL service is redactable |
| DLE\_Amount | 4, 5, 20, 100 | The required number of DLE peer nodes for the PDL service |
| Resource Property | UE\_Participation\_Allow | Yes or No | Whether a UE is allowed to participate the PDL service as a DLE peer |
| UE\_List | {SUPI 1、SUPI 2、SUPI 3} | The list specifying the UEs participating as DLE peers |
| UE\_Policy\_Map | {SUPI1: Policy 1, SUPI12: Policy 2, SUPI3: Policy3} | A map of key value pairs specifying the participating type and policies of all UEs. For example, UE1 shall be static and participate as a full/micro/client DLE |
| gNB\_Participation\_Allow | Yes or No | Whether or not a base station is allowed to participate the PDL service as DLE peers |
| gNB\_List | {gNB\_ID1, gNB\_ID2, gNB\_ID3, …} | The list specifying the gNB participating as DLE peers |
| gNB\_Policy\_Map | {gNB\_ID1: Policy1, gNB\_ID2: Policy 2, gNB\_ID3: Policy 3, …} | A map of key value pairs specifying the participating type and policies of all gNBs. For example, gNB\_ID1 shall be static and participate as a full/micro/client DLE |
| NF\_Participation\_Allow | Yes or No | Whether or not a NF is allowed to participate the PDL service as DLE peers |
| NF\_List | {NF\_ID1, NF\_ID2, NF \_ID3, …} | The list specifying the NFs participating as DLE peers |
| NF\_Policy\_Map | { NF \_ID1: Policy1, NF \_ID2: Policy 2, NF \_ID3: Policy 3, …} | A map of key value pairs specifying the participating type and policies of all NFs. For example, NF\_ID1 shall be static and participate as a full/micro/client DLE |
| … | … | … | … |

### 8.1.2 DLE Instantiation



1. A network resource request is sent to DLAF. This request can be from the network operator where a resource scheduling, re-scheduling and/or planning are needed; this request can be from another CP NF where a dynamic network resource adaption is needed.
2. DLAF sends a network resource request to Network Manager that is responsible for network resource allocation. This request shall include the information about the number of DLE instances needed, the specifications of the required DLE instances and so on.
3. Network Manager analyses the network resource request and decides a deployment plan. The Network Manager shall identify the feasible domains where the request can be accommodated according to the requested resource amount from DLAF.
4. Network Manager sends individual instantiation requests to specific network domains where the DLE instances will be instantiated. Each instantiation request contains the specification of a DLE instance for deployment.
5. Domain Resource Manager instantiates DLE instances. According to the specification provided by the Network Manager, each Domain Resource Manager creates and deploys the request DLE instances with its local resource.
6. Domain Resource Manager sends a response to Network Manager with the instantiation status. Each Domain Resource Manager reports the creation and deployment status of the requested DLE instances with the profile information of each DLE instance such as identifier, operation status and so on.
7. Network Manager confirms the network resource request status to DLAF. Once the Network Manager gets the status report from each Domain Resource Manager about the requested DLE instances, the Network Manager informs the requesting DLAF by sending a response message to the DLAF.
8. [Optional] Domain Resource Manager may directly confirm the creation and deployment status to the DLAF.

### 8.1.3 PDL Service Deployment

Given a PDL service provisioning request, the procedure to deploy this PDL service in a telecom network is illustrated below:



1. DLAF analyses the PDL service description information (e.g., a data profile with the one or multiple fields in clause 8.1.1) and determines the required software libraries needed on DLE instances for provisioning the requested PDL service.
2. DLAF checks with DLRF for the required software libraries.
3. DLAF sends software library configuration profile to every DLE instance that will be part of the PDL service network. A configuration profile for the PDL service contains at least the following fields:
* Peer-to-Peer protocols
* Distributed consensus: PoW, PoS, DPoS, PBFT and so on
* Incentive mechanism
* Wallet type such as digital currency and/or cryptocurrency
* Ledger policies: Redactability, privacy protection and so on
* Cryptography algorithms: Hash algorithm, encryption algorithms for transaction and/or ledger data
* Smart Contract privilege: only defined smart contracts or open to third party
* Size of DLEs
* Others
1. [Optional] DLE retrieves the missing software libraries from DLRF if needed
2. DLE sends a response to DLAF including the status of the software configuration.
3. DLAF analyses the PDL service description information (e.g., a data profile with the one or multiple fields in clause 8.1.1) and determines the PDL service network configurations for all DLE instances
4. DLAF sends PDL service network configuration profile to each DLE instance.
5. DLE instance local configures the PDL service parameters for bootstrapping the PDL service network.
6. DLE instance sends a response to DLAF with the status of PDL service network configuration

### 8.1.4 PDL Service Update



1. DLAF analyzes the DLE update request and derive DLE update configurations. DLAF identifies the new requirements on DLEs, given a PDL service change request. The PDL service change request can be triggered by the owner or the user of the PDL service.
2. DLAF checks the availability of the software libraries. With the identified new requirements, DLAF interacts with DLRF and checks if the required new software libraries are available from there. DLRF will confirm DLAF the availability of the software libraries needed.
3. DLAF sends a software update request to DLEs. DLAF informs every DLE with the update configuration information where the required software updates are listed.
4. [Optional] DLE retrieves new software libraries from DLRF. According to the update request, each DLE retrieves the required software updates from DLRF.
5. DLE sends software update response to DLAF. Each DLE that has finished the software update confirms the status of the update task to DLAF.
6. If PDL service network also needs update, DLAF analyzes the PDL service change request and derives PDL network update configurations.
7. DLAF sends PDL network update request to individual DLEs. DLAF sends a PDL network configuration update request to each DLE where its local PDL network configuration has to be changed. This configuration updates can include new network parameters such as peer DLE list (addresses), network interfaces and so on.
8. DLE applies the new PDL network parameters. Each informed DLE updates its local PDL service network configurations. According to the new parameters provided by the DLAF, the configuration updates may include new peer DLE addresses, new network interface parameters and so on.
9. DLE sends a PDL network update response to DLAF. Once the DLE applies the new network parameters to its local configuration, DLE sends a confirmation with the status of the configuration update to DLAF.

### 8.1.5 PDL Service Termination



1. DLAF identifies the PDL network where the PDL service needs to be terminated. DLAF identifies the PDL network where the targeted PDL service locates with a PDL service ID allocated when the PDL service was created before. With this PDL service ID, DLAF locates the specific DLE instances running the PDL service.
2. DLAF sends termination command to the PDL network. With the identified DLE elements, which together form the PDL network for the PDL service, DLAF sends termination signal to each DLE instance to inform them turn down a PDL service with its PDL service ID.
3. DLE terminates its local PDL service logics and network connections to peer DLEs. Every DLE instance takes the PDL service ID to determine the involved PDL service processing logics, and terminate the local processes; meanwhile, informs the peer DLEs its termination.
4. [Optional] If the termination signal contains a parameter indicating offline ledger data storage, the DLE instance shall transfer the local ledger data to DLDSM. The offline ledger storing should be secured with proper access policies to clearly define at least the following properties:
	* Whether or not the offline ledger data can be modified
	* Who is allowed to access the offline ledger data
	* The lifetime of the offline ledger data. For example, the offline ledger data can be deleted completely after a certain period of them
5. [Optional] DLDSM sends a response back to the DLE instance. After the offline ledger data is transferred and store, the DLDSM confirms to the DLE instance with the offline storing status.
6. DLE sends a response to DLAF with the status of the PDL service termination. Once the termination process is completely done, the DLE sends a confirmation response to the DLAF to inform that the specified PDL service has be terminated on the DLE.

### 8.1.6 DLE Redaction Capability Provisioning

A DLE shall be provisioned with blockchain redaction capabilities, which allow the DLE to issue ledger redaction operation (e.g., to update an existing transaction in distributed ledgers, to update an existing block in distributed ledgers). Ledger redaction capabilities to be provisioned to the DLE shall be authorized and granted by the DLGF.

**Figure 8.2.10-1** illustrates a procedure for provisioning ledger redaction capability to DLE-A. In other words, with this procedure, DLE-A will be able to issue ledger redaction operations to modify distributed ledgers of a target native wireless ledger system.



Figure 6. Ledger Redaction Capability Provisioning

The procedure in **Figure 6** consists of the following steps:

1. The DLGF selects DLE-A and determines that DLE-A can issue ledger redaction operations to a target native wireless ledger system. For this purpose, the DLGF retrieves necessary DLE-A’s information from a DLRF. Alternatively, DLE-A sends a Ledger Redaction Capability Request to the DLAF, which will forward the Ledger Redaction Capability Request to the DLGF; this request contains:
	* The requested ledger redaction capability (i.e., DL-RD-Cap-Req).
	* The identifier and/or the blockchain address of DLE-A (DLE-A-ID).
	* T The identifier of the target native wireless ledger which the requested redaction capabilities will be applied to (Native-DL-ID).

According to DL-RD-Cap-Req, the DLGF grants ledger redaction capabilities to DLE-A in the following step

1. The DLGF grants some ledger redaction capabilities (DL-RD-Cap-Granted) to DLE-A. DL-RD-Cap-Granted specifies the following information:
	* DL-RD-Issuer: Indicate the identifier of the ledger redaction issuer. For this case, DL-RD-Issuer is set to the identifier of DLE-A (DLE-A-ID).
	* DL-RD-Mode: Indicate the ledger redaction mode, which could be:
		+ Direct Redaction – The ledger redaction issuer performs or sends redaction operations directly to the target native wireless ledger system via itself or other DLEs (e.g., DLE-B in the figure).
		+ Indirect Redaction – Each ledger redaction operation from the ledger redaction issuer first needs to be send to and be authorized by the DLAF; then, the DLAF forwards the authorized ledger redaction operation to the target native wireless ledger on behalf of the ledger redaction issuer.
	* Native-DL-ID-by-RD: The identifier of the target native wireless ledger where redaction operations will be sent to or which distributed ledgers will be redacted.
	* DL-RD-Scope: Indicate the scope of ledger redaction (e.g., only certain transactions can be modified, only certain blocks can be modified, etc.)
	* Native-DL-ID-for-RD: The identifier of a native wireless ledger which is used to store the history of ledger redaction operations.
2. The DLGF generates ledger redaction key material (DL-RD-Key-Materials), for example, to derive a ledger redaction key (KDLRD) according to a ledger redaction key scheme (DL-RD-Key-Scheme). DL-RD-Key-Materials contains DL-RD-Key-Scheme, KDLRD, and DLE-A-ID. DL-RD-Cap-Granted is added with a reference to DL-RD-Key-Materials. DLE-A will use the same DL-RD-Key-Scheme in step 6 to generate the same KDLRD.
3. The DLGF signs DL-RD-Cap-Granted and DL-RD-Key-Materials. Then, the DLGF sends signed DL-RD-Cap-Granted and DL-RD-Key-Materials to the DLAF.
4. The DLAF stores DL-RD-Cap-Granted and DL-RD-Key-Materials locally. The DLAF sends DL-RD-Cap-Granted and DL-RD-Key-Materials without KDLRD to DLE-A.
5. DLE-A receives the notification from step 5. DLE-A first verifies the signature contained in DL-RD-Cap-Granted and DL-RD-Key-Materials and stores both locally after their signatures are verified. DLE-A uses DL-RD-Key-Scheme contained in DL-RD-Key-Materials to derive the same KDLRD as the DLGF did in step 3. Then, DLE-A sends a confirmation back to the DLAF.
6. The DLAF sends DL-RD-Cap-Granted to DLE-B.
7. DLE-B creates a new transaction containing DL-RD-Cap-Granted and sends the new transaction to the target native wireless ledger as denoted by Native-DL-ID-by-RD.
8. DLE-B sends a confirmation to the DLAF.
9. The DLAF receives the confirmation from DLE-B and sends another confirmation to the DLGF.

## 8.2 Information Exposure Procedures

### 8.2.1 DLE Information Exposure

DLAF can expose information to DLE. This information can include services of DLAF itself, PDL capabilities that can be installed, the number of PDL services currently running, blockchain functions, and the number of nodes contained in the PDL service. DLAF can also expose information to another DLAF. The information can include the number of DLE instances in the network, the number of nodes, the number of PDL services, blockchain capabilities, etc.

Event: periodic notification, DLAF performs operations on the blockchain (such as updating the blockchain) and DLE (such as initializing the DLE), configures the DLDSM and DLRF, and updates the blockchain profile.

#### 8.2.1.1 DLE Direct Exposure



1. DLAF sends a subscription request to a DLE by using Ndle\_InfoExposure\_Subscribe service on DLE. The parameters of the request include event parameters, subscription duration, and maximum number of reports.
2. The DLE determines if the requesting DLAF is allowed to send such a request. This can be done by verifying the identity of the DLE instance; and this may involve other NFs (e.g., AUSF) to finish the authentication. If yes, the DLE adds the DLAF into its subscription list.
3. The DLE notifies the DLAF that the subscription is successful by using Ndle\_InfoExposure\_Subscribe response service with the status of the subscription.
4. When the subscribed event(s) occur, the DLE notifies corresponding DLAF with Ndle\_InfoExposure\_Notify service with the event data.

#### 8.2.1.2 DLE Indirect Exposure



1. A NF consumer sends a subscription/unsubscription request NEF by using Nnef\_EventExposure\_Subscribe / Unsubscribe Request. The request carries the subscribed event IDs, subscription duration, and maximum number of reports.
2. NEF adds the requesting NF into the subscription list and sends a subscription request to the targeted DLE by using Ndle\_InfoExposure\_Subscription request service with the parameters NEF receives from the requesting NF in the first step.
3. The DLE determines if the requesting DLAF is allowed to send such a request. This can be done by verifying the identity of the DLE instance; and this may involve other NFs (e.g., AUSF) to finish the authentication. If yes, the DLE adds the DLAF into its subscription list.
4. The DLE notifies the NEF of the subscription success by using Ndle\_InfoExposure\_Subscription response service with the subscription status.
5. When detecting a subscription event, the DLE notifies NEF by using Ndle\_InfoExposure\_Notify service with the subscribed events and data.
6. After NEF receives the notification from the DLE. The NEF further notifies the corresponding NF consumer who is the subscriber with occurred event(s) with event data.

This procedure enables that other NFs can retrieve interested events on the DLE such as selecting a particular DLE as a PDL service proxy, and so on.

### 8.2.2 DLRF Information Exposure

The DLRF can expose information to the DLAF e.g., available PDL software library installation packages. The trigger event may be periodic notification, updating the blockchain installation package.



1. DLAF subscribes to the DLRF by sending a subscription request with using Ndlrf\_InfoExposure\_Subscribe service. The request message carries the subscribed event parameters, subscription duration, and maximum number of reports.
2. The DLRF determines if the requesting DLAF is allowed to send such a request. This can be done by verifying the identity of the DLE instance; and this may involve other NFs (e.g., AUSF) to finish the authentication. If yes, the DLRF adds the DLAF into its subscription list.
3. The DLRF notifies the NEF of the subscription success by using Ndle\_InfoExposure\_Subscription response service with the subscription status.
4. When the subscribed event(s) occur, the DLRF notifies corresponding DLAF with Ndle\_InfoExposure\_Notify service with the event data.

The DLAF determines whether to install, update, or delete blockchain capabilities for the DLE based on the information exposed by the DLRF.

### 8.2.3 PDL Service Information Exposure

DLAF can expose information to DLE, including: DLAF-supported services, blockchain capabilities that can be installed, the number of blockchains currently running, blockchain functions, and the number of nodes contained in the blockchain. DLAFs can expose information to other DLAFs, including: the number of DLEs in the network, the number of nodes, the number of blockchains, blockchain capabilities, etc.

Event: periodic notification, DLAF performs operations on the blockchain (such as updating the blockchain) and DLE (such as initializing the DLE), configures the DLDSM and DLRF, and updates the blockchain profile.

#### 8.2.3.1 PDL service internal exposure



1. A DLE or DLAF-a sends a subscription request to DLAF-b by using Ndlaf\_EventExposure\_Subscribe request service. The subscription request carries the subscribed event parameters, subscription duration, and maximum number of reports.
2. The DLAF-b determines if the requesting DLAF is allowed to send such a request. This can be done by verifying the identity of the DLE instance; and this may involve other NFs (e.g., AUSF) to finish the authentication. If yes, the DLRF adds the DLAF into its subscription list.
3. DLAF-b notifies the NEF of the subscription success by using Ndlaf\_EventExposure\_Subscription response service with the subscription status.
4. DLAF-b sends a subscription request a proxy DLE in a PDL service network by using Ndle\_InfoExposure\_Subscribe request service. The parameters included in the request is the same as the parameter the DLAF-b received from the DLAF-a /DLE in the first step.
5. When the subscribed event(s) occur, the DLE proxy notifies DLAF-b with Ndle\_InfoExposure\_Notify service with the event data
6. After NEF receives the notification from the DLE. The DLAF-b further notifies the corresponding NF consumer (DLAF-a/DLE in the first step) who is the subscriber with occurred event(s) with event data.

Subsequently, the DLE may determine, based on the information, whether to request to join or exit a blockchain. DLAFs can assist DLEs in moving to other network domains and determine whether to switch DLEs to other blockchains based on the information exposed by other DLAFs. The high-level DLAF may manage and configure the blockchain capability of the subdomain based on the blockchain status of the exposed supervision subdomain of the sub DLAF, install/delete/update the blockchain capability for the subdomain DLE, and establish/delete/update the blockchain for the subdomain.

#### 8.2.3.2 PDL service external exposure via NEF



1. The NEF sends a subscription request message to the DLAF. The message carries the subscribed event parameters, subscription duration, and maximum number of reports. If the NEF is not the NF that receives the subscription information, the message also carries the ID/IP address of the NF that receives the subscription content.

2. The DLAF determines whether to provide subscription to the NF or NEF. If yes, the DLAF records the NF or NEF ID/IP and the subscription content.

3. The DLAF notifies the NEF of the subscription success.

4. - 5. Blockchain nodes auto-update (e.g., DLE automatically joins or exits the blockchain without going through DLAF), the DLAF updates the blockchain profile.

5. When detecting a subscription event, the DLAF sends the subscribed content to the NF.

6. When the DLAF detects that an event related to the subscription event occurs (for example, link establishment or deletion), the DLAF notifies the NEF of the event.

The DLAF located in the core network may expose information to other network elements in the core network through the NEF. Other networks learn a status of the blockchain by using the DLAF, and may further determine whether to apply the blockchain, join the blockchain, and the like.

### 8.2.4 DLDSM Information Exposure

DLDSM can expose information to DLAF, including: current database storage conditions, such as the size of the stored data, the remaining storage space, etc., as well as the stored blockchain data, such as the blockchain from which the data comes, and the storage time. The event may include periodic notification, adding blockchain data, updating blockchain data, deleting blockchain data, and the like. DLAF accordingly allocates different DLDSMs to different blockchains.



1. The DLAF sends a SUBSCRIBE message to the DLDSM. The SUBSCRIBE message carries the subscribed event parameters, subscription duration, and maximum number of reports.

2. The DLDSM determines whether to provide the subscription to the DLAF. If yes, the DLDSM records the DLAF ID/IP and the subscription content.

3. The DLDSM notifies the DLAF that the subscription is successful.

4. - 5. DLE uploads local storage data to DLDSM, and DLDSM updates blockchain database data.

6. When detecting a subscription event, the DLDSM sends the subscribed content to the DLAF.

## 8.3 Mobility Management Procedures

### 8.3.1 General Information

In a PDL service network, a DLE instance can be deployed on a resource node with mobility. The mobility can be caused by a UE moving from one place to another place; a virtualized network node is migrated from one domain to another domain; or a network node loses connectivity to its peer nodes. In these situations, the PDL service architecture requires the capability to handle the mobility event of a DLE instance in order to guarantee the service continuity of a PDL service.

### 8.3.2 PDL Service Network Scale-Up

#### 8.3.2.1 A new DLE joining in via DLAF



#### 8.3.2.2 A new DLE joining in via a peer DLE



### 8.3.3 PDL Service Network Scale-Down

#### 8.3.3.1 Direct DLE leaving a PDL service network



#### 8.3.3.2 Indirect DLE Leaving a PDL Service network



## 8.4 PDL Service Address Management Procedures

TBD …

# 9. Integration scenarios in a telecom network with PDL service provisioning capability

## 9.1 General

A typical telecom network architecture is assumed as the foundation of the proposed enhancements for PDL service provisioning. Specifically, we consider a 5G network architecture consisting of different network segments such as UE, RAN, transport network, core network and data network (Internet).

Generally, given the assumed telecom network architecture, there are two new NFs that will be added. The first one is a control function, which is called Ledger Anchor Function (DLAF). Another function is a BC Enabler Function, which is the main function to realize a PDL service with the resource within the telecom network. DLAF controls BC Enabler function to provision PDL services over the telecom network infrastructure.

As a nationwide infrastructure, telecom network already becomes a fundamental service provisioning platform for various service applications, across basic mobile Internet connectivity to compute-oriented tasks for both mobile users and over-the-top (OTT) service providers. Thanks to the distributed, reliable and high availability natures, ICT infrastructure shows unique benefits PDL service provisioning as well.

However, different from normal (mobile/OTT) applications, a PDL service is in a form of a blockchain network consisting of a set of distributed peer nodes interconnecting each other. As a result, PDL service provisioning within telecom network is a non-trivial task, because an operator must consider how a blockchain network can be instantiated within the telecom network infrastructure given the specific requirements of a PDL service as well as the resource constraints of the telecom network in itself.

The new requirements on native PDL service provisioning drives a need of architecture enhancement of the telecom network itself. A native PDL service provisioning is based on an end-to-end (E2E) telecom network infrastructure in a dynamic environment; in addition, a native PDL service provisioning can serve both as an OTT and as telecom operator’s services. The specified enhancements (via extending architectural and signalling aspects) integrate the blockchain capability as part of the native/fundamental features of the telecom network.

## 9.2 PDL service provisioning capability native in telecom network

In the network architecture of the telecom network, DLAF can be used as the network function of the core network, manage and control the BC enabler in the network, and schedule and configure the underlying blockchain capabilities to create and maintain the blockchain according to the needs of the business and administrators. BC enabler is an end-to-end capability in the telecom network, which can be combined with UE, base station, NF, or deployed in the telecom network as an independent blockchain all-in-one node, and nodes such as UE, base station, NF and other nodes can only be used as users of the chain.



Figure 7. A native integration with PDL service provisioning capability in a telecom network

## 9.3 PDL service provisioning capability as a network slicing in telecom network

The vertical hierarchical architecture of endogenous blockchain in telecom networks is based on the network deployment of multi-layer DLAF for blockchain management. Sub DLAF: A sub-DLAF of the domain, the creation and management of BC capabilities within the authority domain, responsible for the deployment, creation, access control, etc. of the local ledger.

DLAF vertical layering architecture is based on the network for DLAF layered deployment, each Sub DLAF manages the subdomain blockchain, and the subdomain can be divided according to the RAN network and the CN network. Sub DLAFs are managed directly by the parent DLAF, while all level DLAFs are overall scheduled by the top-level DLAF located in the core network. The top-level DLAF can directly parse the blockchain requirements and issue the decomposed requirements to the subordinate DLAF, and can also analyze the decomposed blockchain requirements after distributed negotiation with other top-level DLAFs and then send them to the subordinate DLAF.

## 9.4 Deployment Considerations of PDL Service Functions

### 9.4.1 DLAF Deployment Options

Logically, DLAF is an NF in the core network. However, its instantiation can be either centralized, distributed with or without a hierarchy of multiple layers or mixed options.

A centralized option means that DLAF instances all locate at a center office such as in the operator’s central service room. For example, a central server room can be a telecom cloud service platform. In practice, DLAF instances in this deployment option are far from the edge of the telecom network.

A distributed option means that DLAF instances distribute at different domains in the telecom network logically and/or geographically. Each DLAF may manage and control the local PDL service requests. Among the distributed DLAF instances, there should be a synchronization mechanism in order to avoid collisions; in addition, there should also be a coordination mechanism among the distributed DLAF instances when inter-domain PDL service provisioning is needed. In practice, DLAF instances in this deployment option can be deployed closer to the edge of the telecom network.

A hierarchy option is a mixture of a distributed option and a centralized option. This means that there will be a centralized DLAF but with different layers of DLAF instances in a distributed manner. If there is a conflict such as for service provisioning or status asynchrony, the collision will be handled by an DLAF instance at a higher layer. In this option, there exists one or more DLAF instances with central authority to organize, control and manage the DLAF instances at the lower layer.

### 9.4.2 DLE Deployment Options

The placement of a BC Enabler Function can locate at any type of nodes in the telecom network. For example, it can run on a UE, an NF either in control plane, user plane or both. A BC Enabler Function can also be instantiated standalone as an individual function or even a server machine when natively co-locating with other entities does not meet the provisioning requirements. In any instantiating form that a BC Enabler Function can be deployed, the execution mode of the BC Enabler Function cam be one of the modes specified in clause 5.3.

### 9.4.3 DLRF Deployment Options

### 9.4.4 LSDM Deployment Options

# 10. Conclusion

*TBD …*

Annex A (normative or informative):
Title of annex

Annex (informative):
Bibliography

# History

|  |
| --- |
| **Document history** |
| V0.0.0 | 04-2023 | Initial table-of-content (ToC) created |
| V0.0.1 | 04-2023 | Brief contents updated for major clauses |
| V0.0.2 | 06-2023 | Enhance clause 5.1 to 5.3, and add clause 5.4 |
| V0.0.3 | 06-2023 | BCR added |
| V0.0.4 | 07-2023 | BC Enabler instantiation procedure added |
| V0.0.5 | 07-2023 | Extended Toc |
| V0.0.6 | 08-2023 | Restructured the whole content and added function services |
| V0.0.7 | 09-2023 | Further revised the structure and update clauses 4 and 8 |
| V0.0.8 | 10-2023 | Initial integration with contributions from PDL(23)000\_145 and \_145. Main update to clause 4 architecture design. |
| V0.0.9 | 11-2023 | Architecture models are stable. |
| V0.1.0 | 11-2023 | Integrated CR-159 and -160 with the definition of DLGF and a high-level feature for PDL service address management |
| V0.1.1 | 11-2023 | Finish Section 5 and 6; briefly drafted Section 7 |
| V0.1.2 | 12-2023 |  Added clause 8.1.1 |
| V0.1.3 | 01-2024 |  Almost finish Section 7 |
| V0.1.4 | 01-2024 |  Further completed Section 7 and added clause 8.2.2.1 |

*Latest changes made on 2024-01-25*