ETSI GS PDL 024 V1.1.1 (2024-11)

Permissioned Distributed Ledgers (PDL); Architecture enhancements for PDL service provisioning in telecom networks

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-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

Contents

Table of figures

Intellectual Property Rights

Essential patents

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

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

Trademarks

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

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

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.

Executive summary

The present document outlines the architecture enhancements for Permissioned Distributed Ledger (PDL) service provisioning in telecom networks. The present document, produced by the ETSI Industry Specification Group (ISG) for Permissioned Distributed Ledger (PDL), aims to specify the technical solutions necessary for enabling telecom networks to provision various PDL services over their infrastructure. Key aspects of the present document include:

- **PDL Service Provisioning Architecture Model**: The architecture model is designed to accommodate and operate PDL services in next-generation telecom networks, taking into account the constraints of Public Land Mobile Networks (PLMNs) such as geographically segmented network domains and heterogeneous resource capacities.
- **PDL Functions**: The architecture consists of several PDL functions, including the Distributed Ledger Anchor Function (DLAF), Distributed Ledger Repository Function (DLRF), Distributed Ledger Enabler (DLE), Distributed Ledger Data Storage Management (DLDSM), and Distributed Ledger Governance Function (DLGF).

- **High-Level Features**: The present document specifies high-level functionalities and features of the PDL provisioning architecture, including PDL service management, onboarding, connectivity management, security aspects, performance assurance, and information exposure.
- **Procedures for PDL Service Provisioning**: Detailed procedures are provided for PDL service provisioning, including service description, DLE instantiation, service deployment, onboarding, update, termination, and redaction capability provisioning.
- **Integration with Telecom Networks**: The present document discusses various integration options for PDL capabilities with telecom networks, including telecom-native and telecom-connected PDL capabilities, and deployment considerations for PDL functions.
- **Recommendations**: The present document concludes with recommendations for further study on integrating the proposed PDL service provisioning architecture with telecom network architecture to provide PDL-enhanced telecom network services, focusing on signalling protocol design, integration of PDL service procedures with telecom network service procedures, and integration of PDL service signalling with telecom network service protocols.

Introduction

The present document outlines the architecture enhancements for Permissioned Distributed Ledger (PDL) service provisioning in telecom networks. Produced by the ETSI Industry Specification Group (ISG) for Permissioned Distributed Ledger (PDL), the present document aims to specify the technical solutions necessary for enabling telecom networks to provision various PDL services over their infrastructure. The integration of PDL capabilities into telecom networks is driven by the need for secure, reliable, and scalable distributed ledger services that can support a wide range of applications, from basic mobile internet connectivity to compute-oriented tasks for both mobile users and Over-The-Top (OTT) service providers. The proposed enhancements focus on extending the architectural and signalling aspects of telecom networks to integrate distributed ledger capabilities as part of their native features. The present document covers the architecture model for PDL service provisioning, high-level features of the system, detailed descriptions of PDL functions, and procedures for PDL service provisioning. It also discusses various integration options for PDL capabilities with telecom networks, including telecom-native and telecom-connected PDL capabilities, and provides recommendations for further study on integrating the proposed PDL service provisioning architecture with telecom network architecture to provide PDL-enhanced telecom network services.

1 Scope

The present 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 present document 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 enhancing functions);

2) Functionalities of the new functions and/or enhanced 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 in the ETSI docbox.

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] ETSI GS PDL 023 (V1.1.1): "PDL service enablers for Decentralized Identification and Trust Management".

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] "Merkle-tree" from Wikipedia® in English, 10 September 2024. Page Version ID: 1245066542.

[i.2] "Trie" from Wikipedia® in English, 02 September 2024. Page Version ID: 1243621934.

[i.3] 3GPP TS 23.501 (V19.0.0): "3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; System architecture for the 5G System (5GS); Stage 2 (Release 19)".

3 Definition of terms, symbols and abbreviations

3.1 Terms

Void.

3.2 Symbols

Void.

3.3 Abbreviations

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

3GPP	3rd Generation Partnership Project
AF	Application Function
AMF	Access and Mobility Function
AUSF	Authentication Server Function
CMP	Certificate Management Protocol
DAPP	Decentralized APPlication
DLAF	Distributed Ledger Anchor Function
DLDSM	Distributed Ledger Data Storage Management
DLE	Distributed Ledger Enabler
DLGF	Distributed Ledger Governance Function
DLRF	Distributed Ledger Repository Function
DN	Data Network
KPI	Key Performance Indicator
LBO	Local Break Out
NEF	Network Exposure Function
NF	Network Function
OTT	Over The Top
PDU	Packet Data Unit
PLMN	Public Land Mobile Network
SBI	Service Based Interface
SEPP	Security Edge Protection Proxy
SF	Service Function
SMF	Service Management Function
TEE	Trust Execution Environment
UE	User Equipment
UPF	User Plane Function

4 PDL service provisioning architecture model

4.1 General concept

The architecture model for PDL service provisioning is to design the minimum set of PDL functions 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 PDL function design.

- Enable each PDL function and its services to interact with other PDL 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 Certificate Management Protocol (CMP) messages.
- Wherever applicable, define procedures (i.e. the set of interactions between PDL functions) as services, so that their re-use is possible.
- Support capability exposure.

4.2 Architecture reference model

4.2.1 PDL Functions

The PDL service provisioning architecture consists of the following PDL functions:

- Distributed Ledger Anchor Function (DLAF).
- Distributed Ledger Repository Function (DLRF).
- Distributed Ledger Enabler (DLE).
- Distributed Ledger Enabler Service Function (DLE-SF).
- Distributed Ledger Data Storage Management (DLDSM).
- 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 (SBI) are used in the PDL service control and management plane:

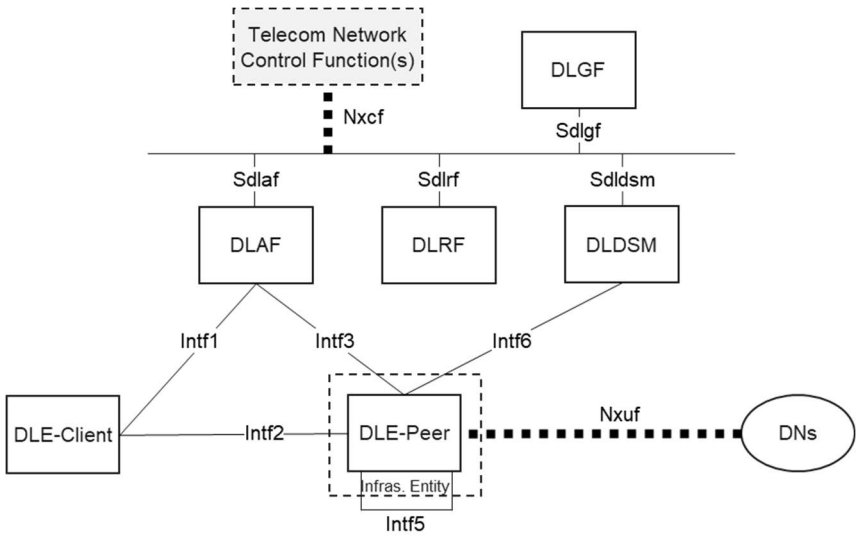


Figure 1: Single-domain PDL service architecture model with SBI in control plane

Figure 2 depicts the single-domain PDL service system architecture with reference points:

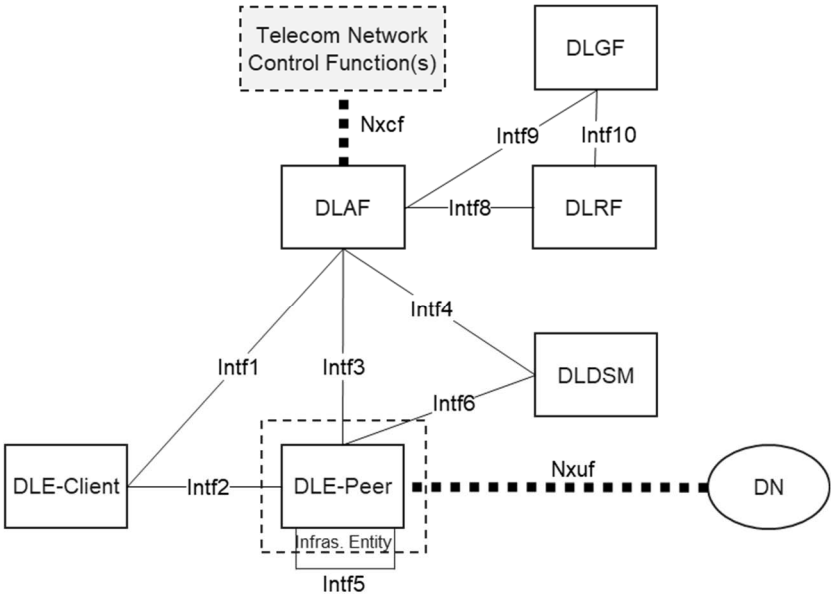


Figure 2: Single-domain PDL service architecture model with reference point representation

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 deployed in a telecom network. The PDL service connects to a Data Network (DN). This PDL service is managed and controlled by a set of PDL functions at the upper part. In addition, the PDL functions 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 function; 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.

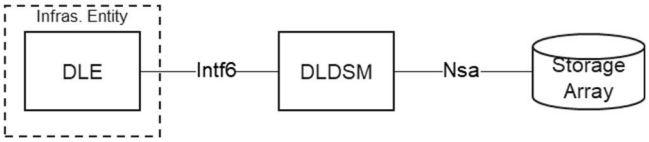
NOTE 3: Another PDL service network can be provisioned in DN. The existing PDL service running on DLEs can access to the other PDL service network via Nxuf interface. This interface can link to a UPF or a direct connection to DN. A PDL service network in DN can run the same PDL service of one consortium or a different PDL service for inter-ledger/blockchain operations. In addition, the other way to inter-work with another PDL service network is via Security Edge Protection Proxy (SEPP), instead of UPF.

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 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 3GPP 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.



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

Figure 3: Architecture for external ledger data storage

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 depicts the architecture for PDL service information exposure:

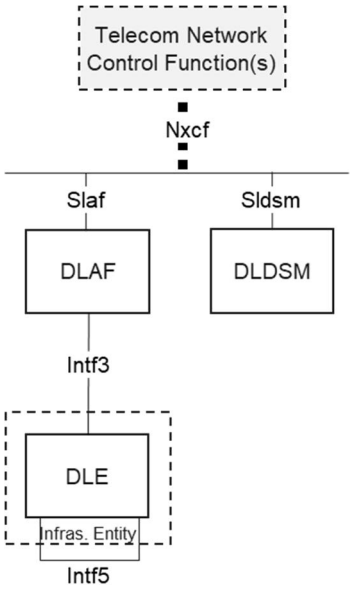


Figure 4: Architecture for PDL service information exposure

The information of a PDL service can be shared internally and externally via an SBI manner. For internal cases, operators may need the service information of the provisioned PDL service for operational purposes such as charging, QoS adaption and so on. For external cases, service providers may also need the service information to determine how to influence and/or interact with the operator for service adaptation and so on.

4.2.5 Architecture to support cross-domain PDL service deployment

Figure 5 depicts the architecture for cross-domain PDL service deployment:

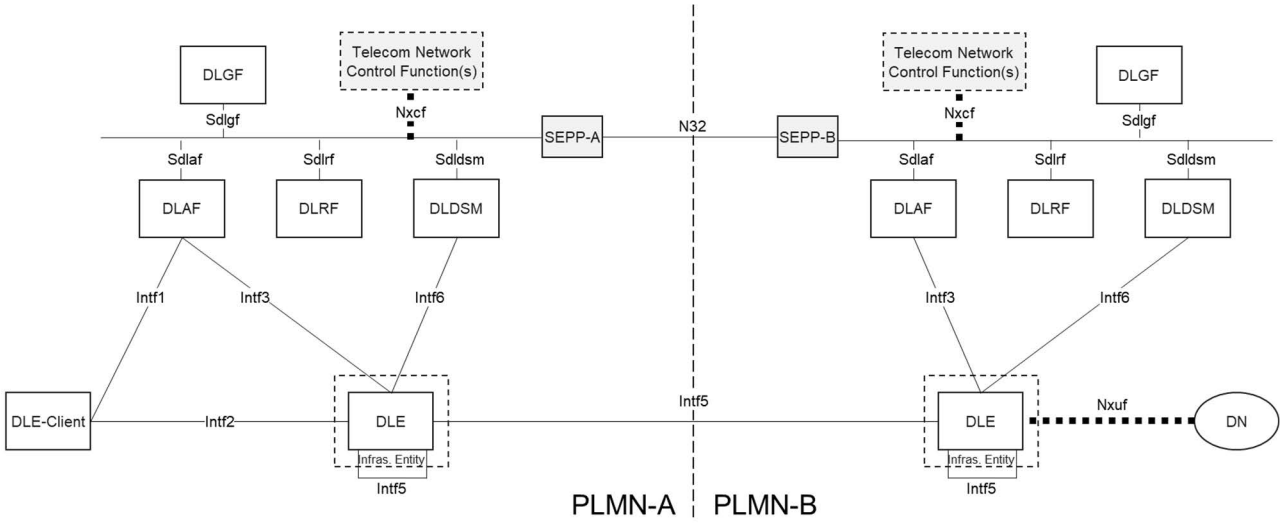


Figure 5: Architecture for cross-domain PDL service provisioning

A PDL service can be deployed across multiple PLMN domains. Different domains can refer different operational domains of one PLMN operator, or different network domains of different PLMN operators where their ownerships can be completely different.

NOTE: A PDL service provisioning is assumed that it is not 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. Temporally extending a PDL service to a visiting PLMN requires much more efforts to build the PDL service network in the visiting PLMN part, which could trigger a lot of overheads in both PLMNs (for signalling and synchronization).

4.2.6 Service-Based Interfaces (SBIs)

The PDL service provisioning system architecture contains the following SBIs:

Sdlaf: SBI of DLAF

Sdlrf: SBI of DLRF

Sdldsm: SBI of DLDSM

Sdlgf: SBI of DLGF

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

Intf9: Reference point between the DLGF and DLAF

Intf10: Reference point between the DLGF and DLRF

Nxcf: A group of interfaces between the DLAF and other Telecom Network Functions

NOTE: The reference point between the DLAF and other telecom network (control) functions reuse the reference points defined in 3GPP TS 23.501 [i.3] for interacting with typical Network Functions (NFs) in a PLMN.

5 High level features of the system

5.1 General

This clause specifies the high-level functionalities and features of the PDL provisioning architecture.

5.2 PDL service management

The PDL service architecture shall support the whole lifecycle management and control of a PDL service from the time the PDL service is requested, its provisioning, deployment and operations, until its termination. In addition, the PDL service architecture shall also support the management of smart contracts intended to be deployed as an application logic of the PDL service. Specifically, PDL service management shall realize the following features:

- Handle and parse the PDL service deployment request.
- Identify network resources feasible for PDL service deployment request.
- Configure network resources with DLE capabilities (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 contracts of a PDL service (e.g. compatibility, validness and threat analysis of a smart contract), which is the responsibility of DLAF and/or DLGF.

5.3 PDL service onboarding

To leverage PDL technology for enabling future trustworthy wireless system, future wireless system needs to integrate PDL capabilities e.g. as new NFs or collocated with existing NFs in the telecom network architecture, referred to as native distributed ledger. For a DLE such as a UE as a DLE-Client to efficiently interact with such a native distributed ledger (e.g. represented by one or more DLEs), the DLE first needs to be properly onboarded to the native distributed ledger and be provisioned with necessary configuration information for interacting with it, referred to as DLE onboarding to native wireless PDL or PDL service onboarding.

PDL service onboarding shall realize the following features:

- Select DLEs that need to be onboarded to a native distributed ledger.
- Determine PDL service address (e.g. a blockchain address) generation scheme for each selected DLE.
- Determine DLAF for each selected DLE.
- Notify the determined PDL service address generation scheme and the determined DLAF to each selected DLE.
- Authenticate the PDL service address of each selected DLE.

- Provision distributed ledger configuration information to each selected DLE.

5.4 PDL service connectivity management

The PDL service architecture shall maintain the connectivity of a DLE to a provisioned PDL service especially under mobility scenarios, which could be a DLE (acting as either a client or a peer) running on a mobile node. The PDL functions 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 is required. Specifically, PDL service connectivity management includes the following features:

- Establish connections for both a DLE-Client to a DLE-Peer and connections among multiple DLE-Peers under the instruction of DLAF and/or DLGF.
- 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/Peer connectivity update with interacting and coordinating with other PDL functions (e.g. existing NFs in the telecom network).

5.5 PDL service security aspect

The PDL service architecture shall handle the security aspect of a PDL service that is requested, provisioned and operated in the telecom network. Specifically, this aspect includes the following features:

- Generate and distribute cryptographical materials for both DLE-Clients 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 integrity for PDL service data including user data (e.g. idfunction and generated ledger data), signalling data between any two entities such as DLE-Client, PDL functions (e.g. DLAF) and DLE-Peers.
- Configure security policies for both DLE-Client and PDL functions (e.g. DLEs, DLRF, DLDSM) with the guidance from governance layer (e.g. DLGF).
- Analyse abnormal/malicious behaviours of a deployed PDL service; if needed, trigger to prescribe corresponding mitigation strategies (by coordinating with other PDL functions.
- Provide Trusted Execution Environment (TEE), if supports from hardware resource are available, for operating smart contract and distributed consensus mechanism.

5.6 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 functions of the telecom network including both 3GPP NFs and the present PDL functions in PDL service architecture, performance assurance includes the following features:

- Collect Key Performance Indicators (KPIs) of a deployed PDL service.
- Analyse 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).

5.7 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 PDL function can subscribe the performance status from DLEs; with the collected information, a PDL function 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.

5.8 PDL service address management

PDL service address is defined as a distributed ledger address (e.g. a blockchain address) that is used in transactions to be sent to distributed ledgers. A transaction has two distributed ledger addresses: a sender address indicating the sender of the transaction and a receiver address indicating the receiver of the transaction.

- DLE-Client needs to have an address on PDL service network in order to access services provided by DLE-Peer (e.g. send a transaction to DLE-Peer).
- A DLE-Peer also has a blockchain address, which is used to send transactions to the PDL service network for control and management purpose.

Those PDL service addresses shall be permissioned as a part of PDL system. PDL service address management shall include the following features.

- PDL service address generation.
- PDL service address authorization and authentication.
- Maintain the mapping between PDL service address and DLE's identifier in wireless system (e.g. 3GPP identifier).

6 PDL function

6.1 General

This clause defines PDL functions 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. DLE has two different types. The first type is acting as a DLE-Client and the second type is acting as a DLE-Peer.

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 the 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 is optional. A node in micro mode acts as a pre-processing node for preparing transactions before submitting the transactions to the PDL service network. The functionalities are split from a lightweight-/full-mode nodes. For example, it can be delegated tasks to accept and verify transactions submitted by the client, compose transactions and package them into micro blocks, and broadcast them to other DLE-Peers. However, a DLE-Peer in micro mode does not participate consensus process while receive consensus results from other DLE-Peers.

- *Lightweight Mode*: A DLE-Peer in lightweight mode contains the features of a micro-node DLE-peer. A DLE-Peer in lightweight mode has the ability to validate micro blocks submitted from micro-mode DLE-peers and participate consensus process. However, a lightweight mode DLE-Peer does not necessarily store full ledger data but partial ledger data.

- *Full Mode*: This mode contains all capabilities of the lightweight mode (also micro mode). In addition, a DLE-Peer in full mode will have to run consensus protocols and store full ledger data with its local storage.

In addition to the mode differences, DLE-Peers in any of the 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 PDL functions.

- *Autonomic*: The topology information, e.g. the addresses of neighbouring DLE-Peers, is autonomically discovered by a DLE-Peer.

6.2.1.4 DLE service functionalities for Telecom network to consume external PDL service

The telecom network can involve scenarios (e.g. a service provisioning to the end-user depends on certain data being managed over an external PDL service) where the core network function(s) that is responsible for taking a decision related to a service request such as authentication needs access to data managed over the external PDL service. In the case of data being managed across different domains (e.g. different service provider domains or multiparty), it is highly likely that the PDL services are managed and operated external to the 3GPP/Telecom network, referred as PDL services infrastructure / framework in Figure 6. Such a PDL service framework is implemented as specified in ETSI GS PDL 012 [1] and ETSI GS PDL 023 [2]. In this scenario, one or more instances of DLE offer the Distributed Ledger Enabler Service Functionality (DLE-SF) (i.e. in case of blockchains it can be blockchain service enabler functionality), to act as proxy and enable the core network function(s) in the telecom network to subscribe and consume the PDL services offered by the PDL services infrastructure / framework to request, fetch, and use appropriate data necessary to process the end-user network service requests and service provisioning. The type of data that is being managed over a PDL and consumed by the core network functions via the DLE-SF depends on the type of Telecom's end-user service use case(s) and it is not in the scope of the present document.

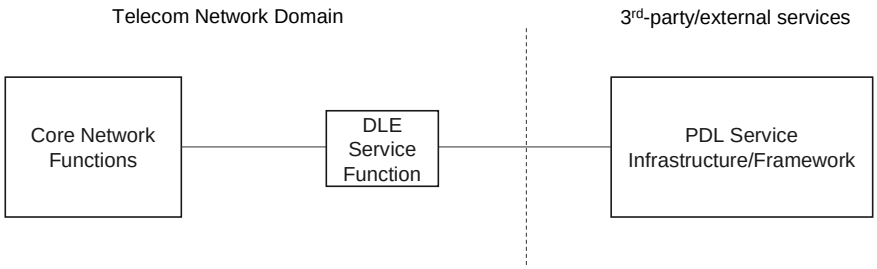


Figure 6: Telecom network consuming PDL service(s) via DLE-SF

6.2.2 DLAF

6.2.2.1 PDL service management

The PDL service management on DLAF includes to:

- Identify feasible network resources in a telecom network infrastructure.
- Prepare network nodes with initial instantiation of DLE (including for both Peer and Client), where major tasks are to:
  - 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 to:
  - 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 to:
  - 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 3GPP NFs and the present PDL functions (e.g. DLGF) in the proposed PDL service provisioning architecture for generating and distributing cryptographical 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 DLE ledger data external 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 DLE-Peer external data 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).

6.2.2.2.3 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 network resource node may not be installed with all software needed to run as a DLE-Peer of a PDL service network, such as a particular distributed consensus protocol stack. 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 distributed ledger 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 Merkle tree [i.1] or Trie [i.2] 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

Distributed 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 function 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 PDL function to coordinate and govern all PDL functions (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 012 [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 behaviour of the PDL platform.
- Governing Function: A function 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 Function 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 external function (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. This function may be done jointly by DLGF and DLAF.
- Coordinate interactions among PDL function entities as defined in the present document from the perspective of governing PDF function entities.
- Coordinate and manage underlying distributed ledger networks such as the management of PDL nodes. This could be done jointly with DLAF.

Coordinate and manage provisioning ledger redaction capabilities to PDL function entities.

7 Function service description

7.1 General

A proposed PDL function offers a capability to authorized consumers. The new PDL functions may offer different capabilities and thus, different functional services to distinct consumers. Each of the functional services offered by the new PDL 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 Functions
Ndlaf_PdlFuncFunction	This service manages other PDL functions in the PDL service provisioning architecture	Activate Deactivate download Add Remove unlock Lock	Any consumer function requires to orchestrate the PDL functions in the proposed PDL service provisioning architecture. For example, the management plane of the telecom network may want to add or remove PDL functions.
Ndlaf_PdlService	This service configures a PDL service with DLE-Peers and manages the whole life circle of a deployed PDL service	Create Stop Update Remove	A consumer that requests to deploy a PDL service in the telecom network.
Ndlaf_InfoExposure	This service provides interfaces to subscribe / unsubscribe event information	Subscribe Unsubscribe Notify Transfer	Any NF or PDL function that is interested a particular event of a deployed PDL service.

Ndlaf_PdlFuncFunction collects a set of services provided by DLAF. The main job of this service interface is to let the consumer PDL function to operate the other PDL functions for PDL service provisioning. This service interface at least contains the following concrete operations to the PDL functions for PDL service:

- Ndlaf_PdlFuncFunction_Activate: This service interface allows a consumer function to activate a specific PDL function for PDL service provisioning. This PDL function is one of the entities proposed in the architecture models, except DLGF.
- Ndlaf_PdlFuncFunction_Deactivate: This service interface allows a consumer function to deactivate a specific PDL function for PDL service provisioning. This PDL function is one of the entities proposed in the architecture models, except DLGF.
- Ndlaf_PdlFuncFunction_Download: This service interface allows a consumer function to download a specific software for a PDL function for PDL service provisioning. This PDL function is one of the entities proposed in the architecture models, except DLGF.
- Ndlaf_PdlFuncFunction_Add: This service interface allows a consumer function to add a new PDL function instance for PDL service provisioning in the telecom network domain. This PDL function is one of the entities proposed in the architecture models, except DLGF.
- Ndlaf_PdlFuncFunction_Remove: This service interface allows a consumer function to remove an existing new PDL function instance for PDL service provisioning in the telecom network domain. This PDL function is one of the entities proposed in the architecture models, except DLGF.

- Ndlaf_PdlFuncFunction_Lock: This service interface allows a consumer function to lock a PDL function instance for PDL service provisioning in the telecom network domain. This will maintain the PDL function but temporally prohibit it actions, which makes this function unavailable. This PDL function is one of the entities proposed in the architecture models, except DLGF.

- Ndlaf_PdlFuncFunction_Unlock: This service interface allows a consumer function to unlock a PDL function instance for PDL service provisioning in the telecom network domain. This will bring back a PDL function that was temporally suspended before and make this function available again. This PDL function 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 function can operate with a PDL service that will be deployed within a telecom network. This service interface at least contains the following concrete operations to the PDL functions for PDL service:

- Ndlaf_PdlService_Create: This interface allows a consumer function 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 function 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 function 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 function 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.

Ndlaf_InfoExposure collects a series of interfaces provided from DLAF where a consumer function can request information about a specific PDL service deployed in the telecom network. This service interface at least contains the following concrete operations:

- Ndlaf_InfoExposure_Subscribe: This interface allows a consumer PDL function to subscribe one or multiple interested events of a PDL service. The interested events indicated by the consumer PDL function will be recorded and corresponding notification will be triggered once the subscribed events occur in the PDL service (e.g. from DLE-Peers).

- Ndlaf_InfoExposure_Unsubscribe: This interface allows a consumer PDL function to unsubscribe one or multiple events of a PDL service that the consumer PDL function is not interested in anymore. The identifier will be removed from the subscriber list maintained by the DLAF.

- Ndlaf_InfoExposure_Notify: This interface allows an event producer triggers an event notification with event data and sends the notification to the subscriber(s) who subscribed the events.

Ndlaf_InfoExposure_Transfer: This interface allows DLAF to transfer one or multiple events together with the event data to another PDL functions. The purposes of the information transfer could be for archive and so on.

7.3 DLE services

The following services are specified for DLE-Peer:

Table 2: List of DLE-Peer Services

Service Name	Description	Service Operation	Consumer Function
Ndle_InfoExposure	This service enables to subscribe and notify event information with data of an interested PDL service on a DLE	Subscribe Unsubscribe Notify	A 3GPP NF and/or a PDL function that are interested in one or multiple events on a DLE-Peer/Client.
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 look up a data record on another DLE and/or synchronize a data record with another DLE	Lookup Sync	DLE
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 Connect	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	Check Activate Deactivate Download Remove	DLE, DLAF

Ndle_InfoExposure collects a series of service interfaces where a consumer function can use to monitor the behaviours/statuses on a DLE instance. This service interface at least contains the following concrete operations to the PDL functions for PDL service:

- Ndle_InfoExposure_Subscribe: This interface allows a consumer function 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 function 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 function 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 function to lookup/read a specific data information from the ledger stored on a DLE instance.
- Ndle_DataTransfer_Sync: This interface allows a consumer function to command a specific DLE instance synchronizing the ledger data with another authorized PDL function.

Ndle_PdlConnection collects a series of service interfaces that allow a consumer function 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 function to command a DLE instance to discover one or multiple neighbour DLE instances in order to form a PDL service DL network structure.
- Ndle_PdlConnection_Connect: This interface allows a consumer function 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 function 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 function to configure the behaviours of a DLE instance. This at least includes the following service interfaces:

- Ndle_Capability_Activate: This interface allows a consumer function 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 function 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 function 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 function 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 Function
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.	Add Delete Update	DLAF
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 function to update the library for realizing PDL services on DLRF including version update, obsolete library removal and so on.	Lookup Insert Delete Update	DLAF, DLE
Ndlrf_InfoExposure	This service allows an authorized consumer function to subscribe/unsubscribe the events from DLRF. This service will be used if the consumer function wants to monitor the software library changes in this PDL function. Whenever a subscribed event occurs, it will be notified to the subscriber(s).	Subscribe Unsubscribe Notify	DLAF, DLE

Ndlrf_Policy provides a series of service interfaces to a consumer function 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 function 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 function will be DLGF.

- Ndlrf_Policy_Delete: This service interface allows a consumer function 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 function will be DLGF.

- Ndlrf_Policy_Update: This service interface allows a consumer function 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 function will be DLGF.

Ndlrf_Library provides a series of service interfaces that allow a consumer function to interact with DLRF for the software library configurations:

- Ndlrf_Library_Lookup: This interface allows the consumer function 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 function 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 function to delete a certain software library that already exists in DLRF.
- Ndlrf_Library_Update: This interface allows the consumer function 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 Function
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, DLE
		Delete	
		Update	
Ndldsm_LgData	This service provides all operations for the ledger data of a DLE including save, update and delete.	Save	
		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	
		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 function 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 function to delete an existing policy on DLDSM.
- Ndldsm_Policy_Update: This service interface allows a consumer function to update an existing policy on DLDSM. For example, a consumer function 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 function 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 function 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 function 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 function to temporally lock a piece of ledger data on DLDSM. Locking the ledger data prohibits data retrieval from a consumer function but the existence of the ledger data is still visible.
- Ndldsm_LgData_Unlock: This interface allows an authorized consumer function to unlock a piece of ledger data on DLDSM that was previously locked.
- Ndldsm_LgData_Lookup: This interface allows an authorized consumer function look up a piece of ledger data with keywords.

Ndldsm_Exposure provides a series of services to expose the ledger data on DLDSM to other functions. It at least contains the follow service interfaces:

- Ndldsm_Exposure_Subscribe: This interface allows an authorized function to subscribe events related to the ledger data status. For example, a function 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 function to unsubscribe events related to the ledger data status that was previously subscribed by the function.
- Ndldsm_Exposure_Notify: This interface allows DLDSM notify an PDL function with an event with event data information. The notified PDL function 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 Function
Ndlgf_DLOnboardingInfo	The service provides interfaces for retrieving distributed ledger configuration information.	Retrieve	DLE, DLAF
Ndlgf_DLAddr	This service provides interfaces for requesting to authorize a PDL service address.	Authorize	DLAF
Ndlgf_DLRedaction	The service provides operations for supporting to provision distributed ledger redaction capability.	Retrieve	DLAF

Ndlgf_DLOnboardingInfo has the following service interface:

- Ndlgf_DLOnboardingInfo_Retrieve: The service interface allows a consumer function (e.g. DLE, DLAF) to retrieve distributed ledger onboarding configuration information for a DLE from DLGF, which in turn can onboard itself to a native blockchain.

Ndlgf_DLAddr has the following service interface:

- Ndlgf_DLAddr_Authorize: The service interface allows a consumer function (e.g. DLAF) to request DLGF to authorize a PDL service address of a DLE.

Ndlgf_DLRedaction has the following service interface:

- Ndlgf_DLRedaction_Retrieve: The service interface allows a consumer function (e.g. DLAF) to retrieve ledger redaction capability for a DLE from DLGF.

7.7 Summary

The parameters of each service interface are not clearly defined as it leaves to implementation options. In addition, the consumer function of each service interface is also exhausted, the present functional functions in the tables are only examples. Other NFs may also invoke the service interfaces here, depending on the concerned scenarios and the need of the whole system design.

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 Table 6.

Table 6: PDL Service Description

Classification	Attribute Name	Description	Example
General Property	Participant ID	The identifiers of all participants forming the PDL service.	ID1, ID2, ID3, …
	Ledger Data Structure	The topology structure organizing ledger transaction data.	Single-chain, multi-chain, or DAG
	Consensus Protocol	The consensus protocol option(s) that shall be used for this PDL service.	PoS, PoW, Raft, PBFT
	Transaction Per Second (TPS)	Required throughput of the PDL service.	100, 500, 1000
	Redactable	Whether the PDL service is redactable.	Yes or No
	DLE_Amount	The required number of DLE peer nodes for the PDL service.	4, 5, 20, 100
Resource Property	UE_Participation_Allow	Whether a UE is allowed to participate the PDL service as a DLE peer.	Yes or No
	UE_List	The list specifying the Ues participating as DLE peers.	{SUPI 1, SUPI 2, SUPI 3}
	UE_Policy_Map	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.	{SUPI1: Policy 1, SUPI12: Policy 2, SUPI3: Policy3}
	gNB_Participation_Allow	Whether or not a base station is allowed to participate the PDL service as DLE peers.	Yes or No
	gNB_List	The list specifying the gNB participating as DLE peers.	{gNB_ID1, gNB_ID2, gNB_ID3, …}
	gNB_Policy_Map	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.	{gNB_ID1: Policy1, gNB_ID2: Policy 2, gNB_ID3: Policy 3, …}
	NF_Participation_Allow	Whether or not a NF is allowed to participate the PDL service as DLE peers.	Yes or No
	NF_List	The list specifying the NFs participating as DLE peers.	{NF_ID1, NF_ID2, NF _ID3, …}
	NF_Policy_Map	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.	{NF _ID1: Policy1, NF _ID2: Policy 2, NF _ID3: Policy 3, …}

NOTE: The table can be extended if more other fields are needed to describe a PDL service.

8.1.2 DLE instantiation

Figure 7 illustrates a procedure where a DLAF interacts with the network manager in order to deploy/instantiate a DLAF instance in the network.

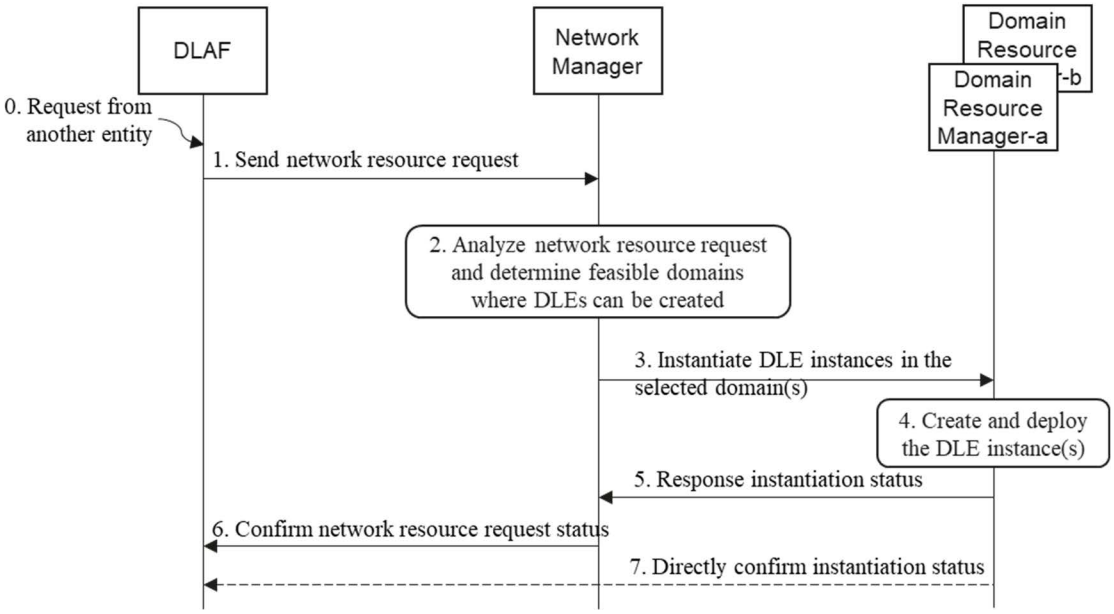


Figure 7: Procedure to instantiate a DLE instance by sending a request to domain resource managers

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.

NOTE: Network manager is a terminology from SDN/NFV. It serves the purpose to realize 3GPP SA5 management plane orchestrating the resource layer. They are not new functions. PDL service provisioning architecture reutilizes the existing capability already defined.

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:

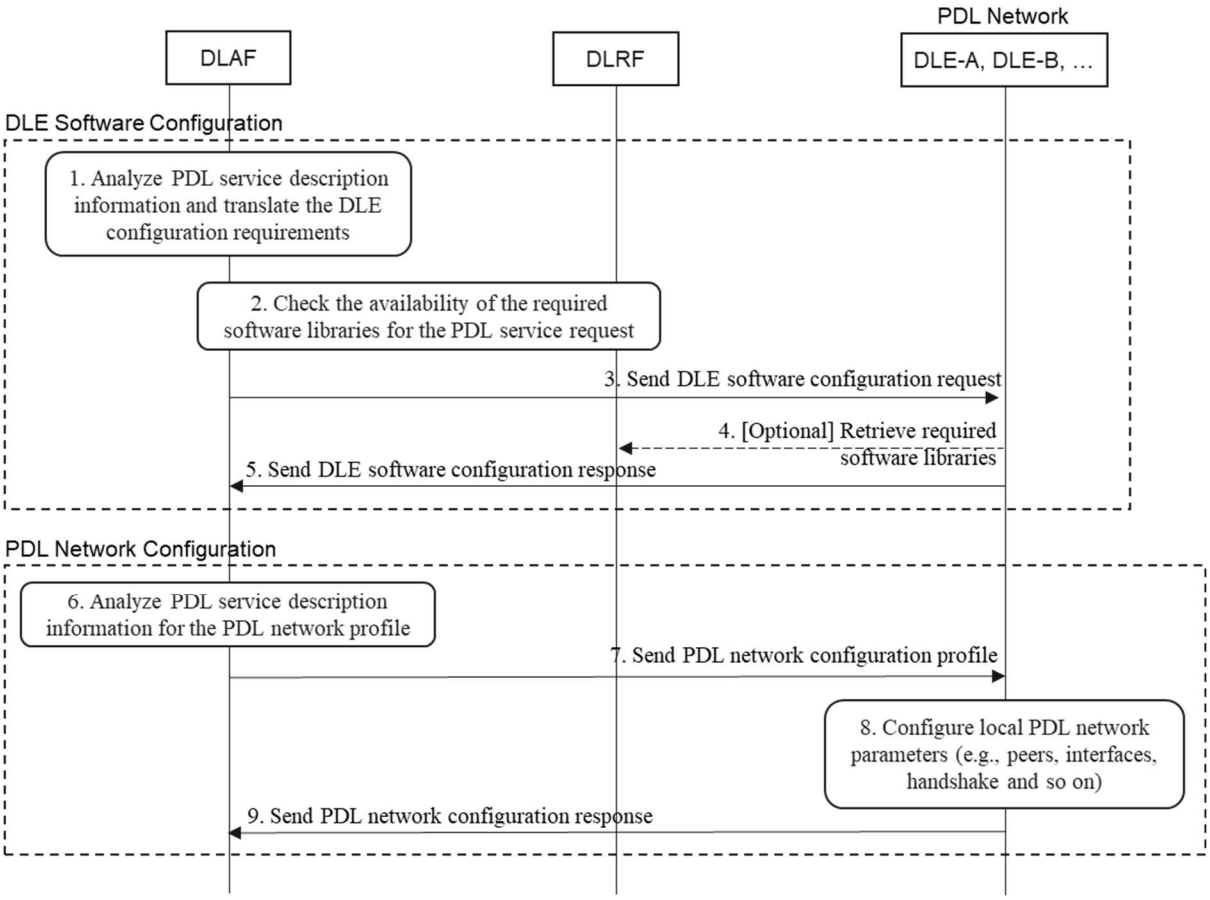


Figure 8: Procedure for provisioning a PDL service by a DLAF to a set of DLE instances

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.

4) [Optional] DLE retrieves the missing software libraries from DLRF if needed.

5) DLE sends a response to DLAF including the status of the software configuration.

6) 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.

7) DLAF sends PDL service network configuration profile to each DLE instance.

8) DLE instance local configures the PDL service parameters for bootstrapping the PDL service network.

9) DLE instance sends a response to DLAF with the status of PDL service network configuration.

8.1.4 PDL service onboarding

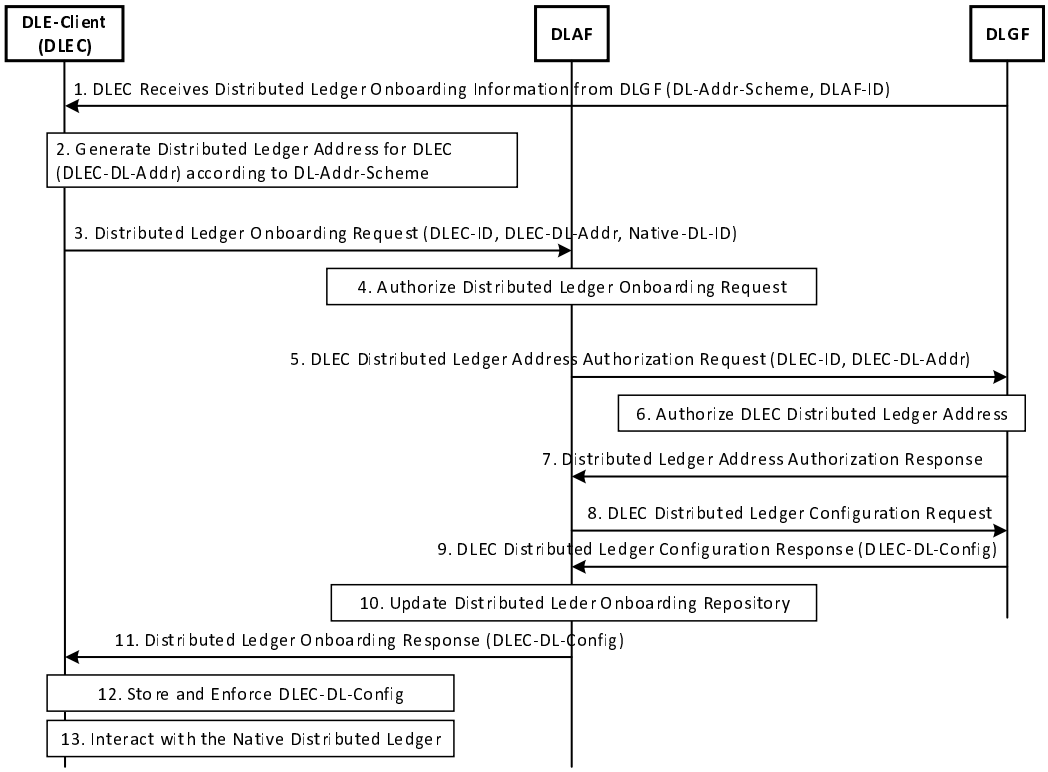


Figure 9: Procedure for PDL service onboarding

Figure 9 illustrates PDL onboarding procedure, which consists of the following steps:

1) DLEC receives distributed ledger onboarding information from DLGF, which may contain:

   a) DL-Addr-Scheme: Indicate the distributed ledger address generation scheme which DLEC shall use to generate its distributed ledger address (i.e. PDL service address).

   b) DLAF-ID: Indicate the identifier of DLAF, which DLEC shall contact in step 3 for sending a distributed ledger onboarding request to.

2) DLEC generates its distributed ledger address (DLEC-DL-Addr) according to DL-Addr-Scheme.

3) DLEC sends a distributed ledger onboarding request to DLAF. This request shall contain the following information:

   a) DLEC-ID: The 3GPP identifier of DLEC.

b) DLEC-DL-Addr: The distributed ledger address of DLEC as generated in step 2.

c) Native-DL-ID: The identifier or the name of native distributed ledger which DLEC is onboarding to.

4) DLAF authorizes the distributed ledger onboarding request received from step 3.

5) DLAF sends a DLEC distributed ledger address authorization request to DLGF. This request shall contain the following information:

a) DLEC-ID: As received in step 3.

b) DLEC-DL-Addr: As received in step 3.

6) DLGF authorizes DLEC distributed ledger address.

7) DLGF sends a distributed ledger address authorization response to DLAF.

8) DLAF sends a DLEC distributed ledger configuration request to DLGF.

9) DLGF sends a DLEC distributed ledger configuration response to DLAF. The response may contain some configurations determined for DLEC (DLEC-DL-Config) such as transaction template, transaction generation rate, a list of DLEs that DLEC can interact with.

10) DLAF creates a new distributed ledger onboarding record for DLEC and adds the record to the maintained distributed ledger onboarding repository.

11) DLAF sends a distributed ledger onboarding response to DLEC. This response shall contain DLEC-DL-Config as received from step 9.

12) DLEC stores the received DLEC-DL-Config locally and enforces it.

13) DLEC interacts with the corresponding native distributed ledger according to DLEC-DL-Config.

8.1.5 PDL service update

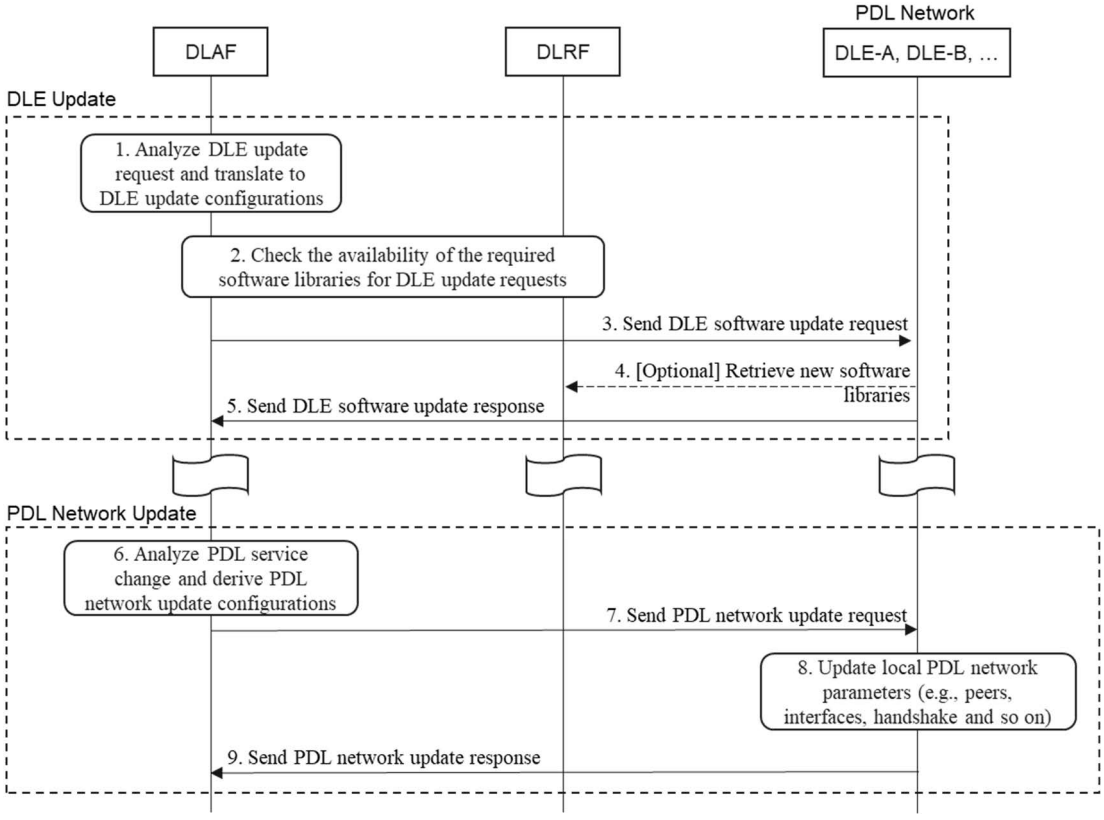


Figure 10: Procedure for update an existing PDL service in the network

1) DLAF analyses 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 analyses 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.

8.1.6 PDL service termination

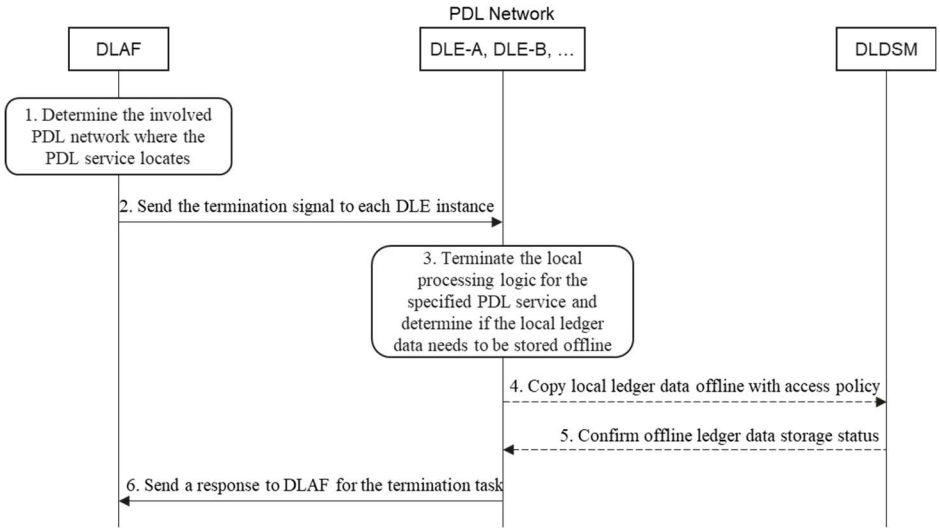


Figure 11: Procedure for terminating an existing PDL service in the network

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.

8.1.7 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 12 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.

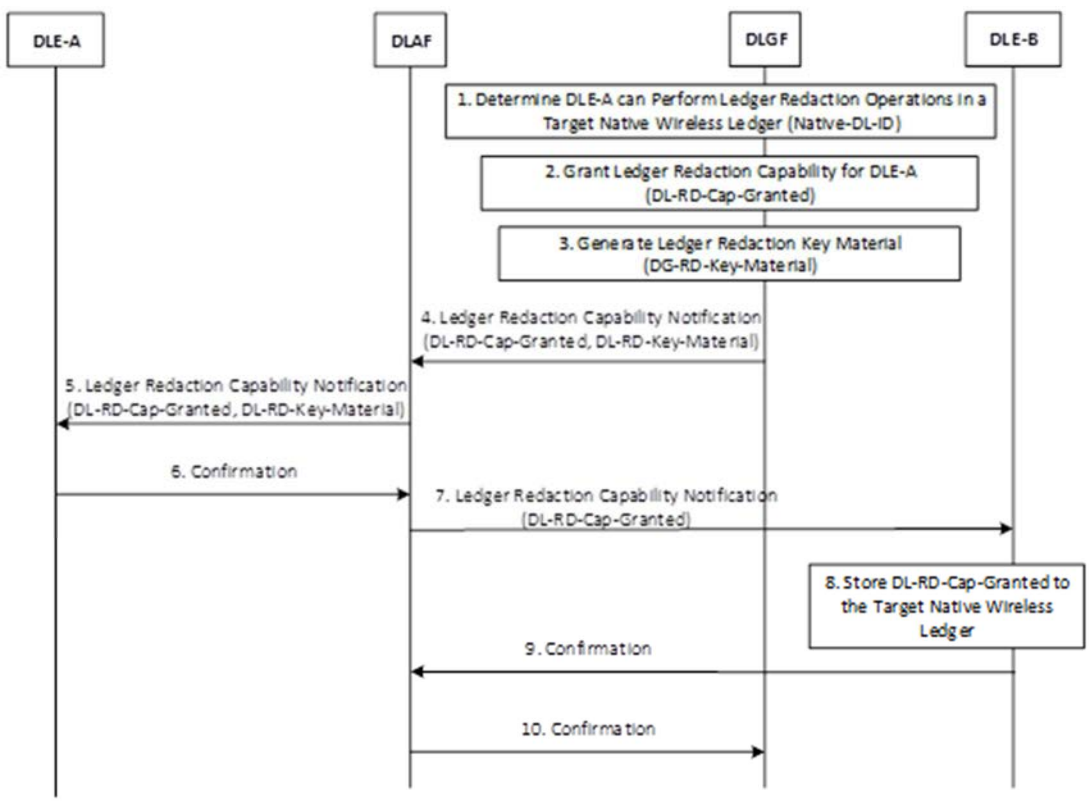


Figure 12: Ledger Redaction Capability Provisioning

The procedure in Figure 12 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:

2) 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 Figure 12).
     - 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.

3) The DLGF generates ledger redaction key material (DL-RD-Key-Materials), for example, to derive a ledger redaction key (K_{DLRD}) according to a ledger redaction key scheme (DL-RD-Key-Scheme). DL-RD-Key-Materials contains DL-RD-Key-Scheme, K_{DLRD}, 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 K_{DLRD}.

4) 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.

5) 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 K_{DLRD} to DLE-A.

6) 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 K_{DLRD} as the DLGF did in step 3. Then, DLE-A sends a confirmation back to the DLAF.

7) The DLAF sends DL-RD-Cap-Granted to DLE-B.

8) 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.

9) DLE-B sends a confirmation to the DLAF.

10) 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

8.2.1.1 General information

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.

The possible events include for example 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.

8.2.1.2 DLE direct exposure

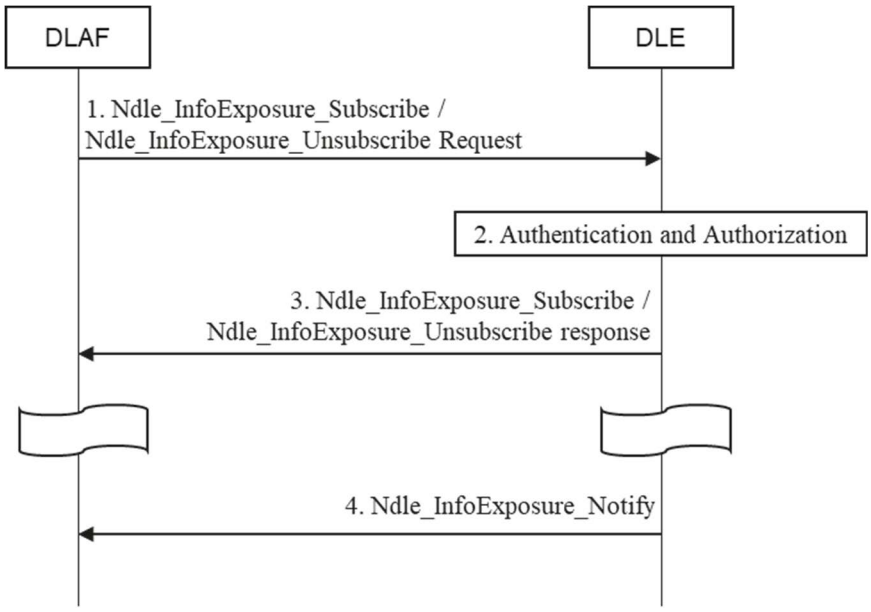


Figure 13: Procedure for DLE information direct exposure

The procedure illustrated in Figure 13 contained the following steps:

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 idfunction of the DLAF; 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.

8.2.1.3 DLE indirect exposure

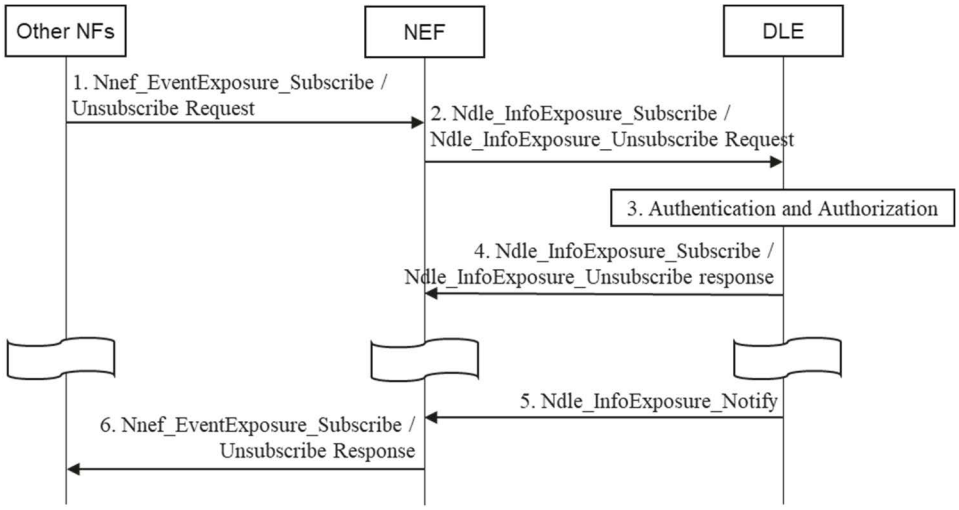


Figure 14: Procedure for DLE information indirect exposure

The procedure illustrated in Figure 14 contains the following steps:

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 NF consumer is allowed to send such a request. This can be done by verifying the idfunction of the NF consumer; 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.

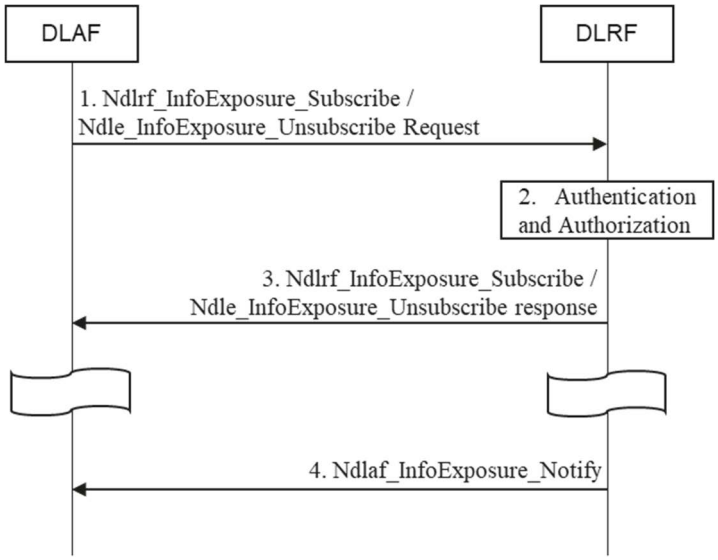


Figure 15: Procedure for DLRF information exposure

The procedure illustrated in Figure 15 contains the following steps:

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 idfunction 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

8.2.3.1 General information

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.

The possible events include for example 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.

8.2.3.2 PDL service internal exposure

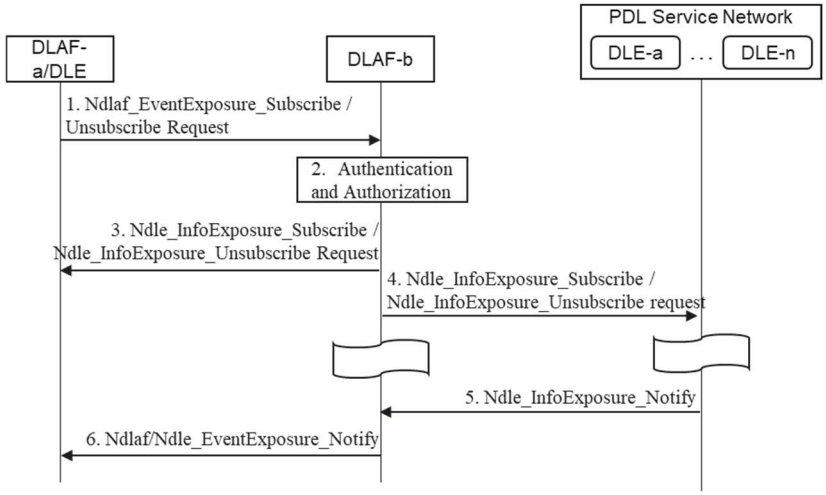


Figure 16: Procedure for PDL service internal exposure to other NFs in the network

The procedure illustrated in Figure 16 contains the following steps:

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 idfunction 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.

8.2.3.3 PDL service external exposure via NEF

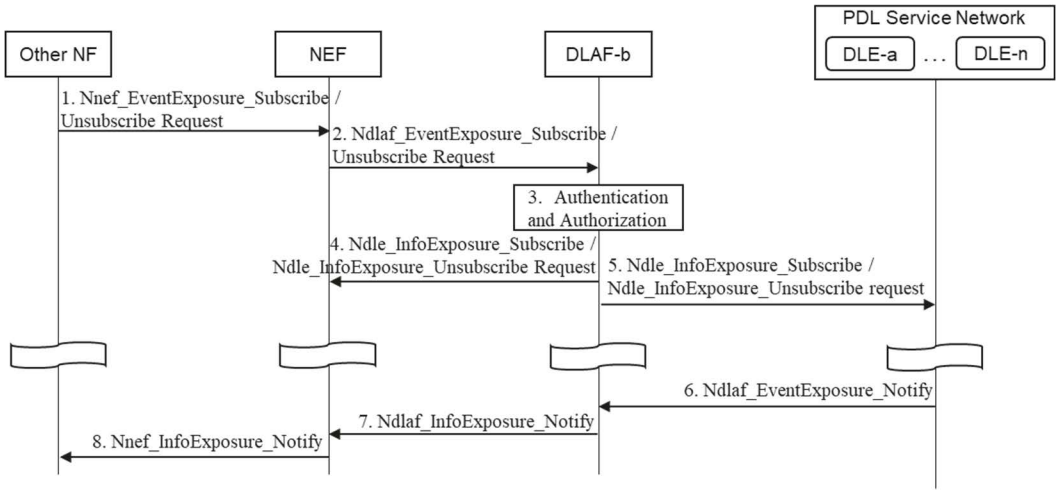


Figure 17: Procedure for PDL service exposure to an external party

The procedure illustrated in Figure 17 contains the following steps:

1) An external NF (e.g. AF) sends a subscription request to NEF.

2) 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.

3) 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.

4) The DLAF notifies the NEF of the subscription success.

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

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

7) 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.

8) NEF will expose the event to corresponding event subscriber NF.

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.

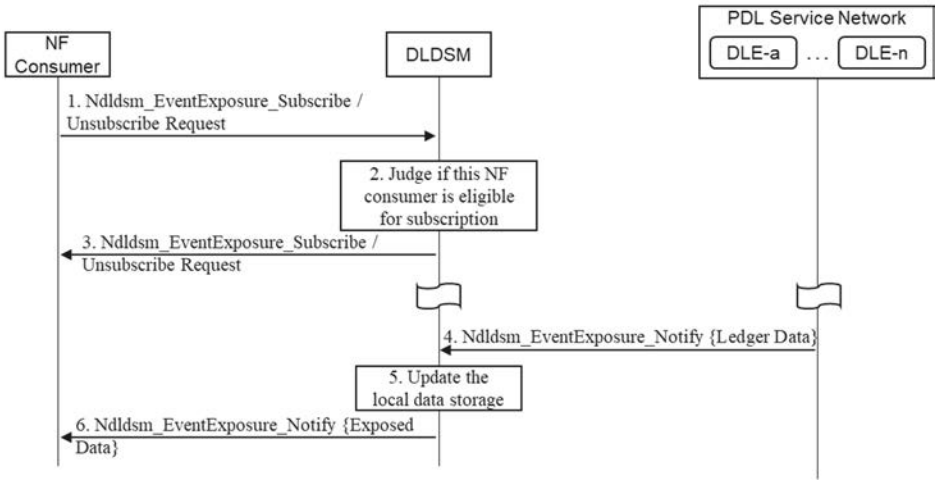


Figure 18: Procedure for DLDSM information exposure

The procedure illustrated in Figure 18 contains the following steps:

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) DLE uploads local storage data to DLDSM, and DLDSM updates blockchain database data.

5) When detecting a subscription event, the DLDSM updates the status of the data

6) DLDSM sends a notification to the corresponding event subscriber NF.

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

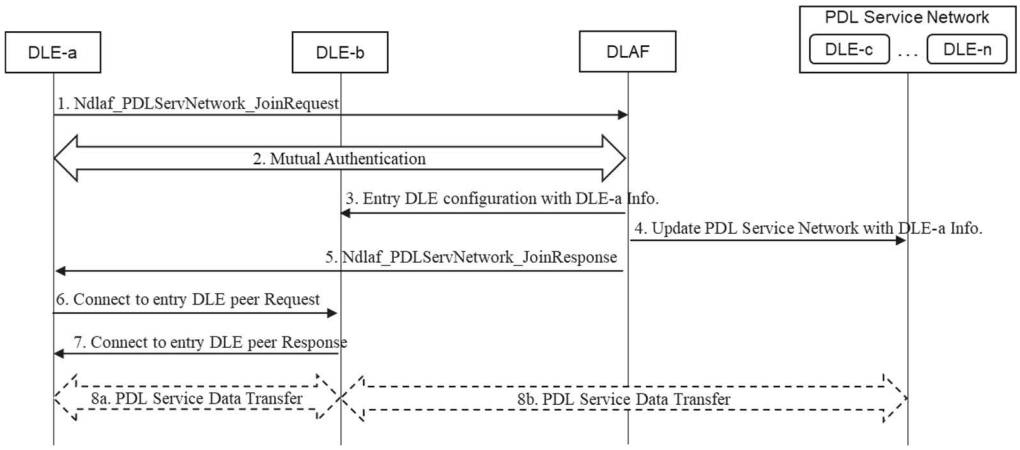


Figure 19: Procedure for a DLE joining a PDL service network via DLAF

The procedure illustrated in Figure 19 contains the following steps:

1) A DLE-a sends a joining request to DLAF.

2) DLAF and the DLE-a proceed mutual authentication procedure.

3) DLAF configures the entry DLE-b for DLE-a where the DLE-b is a DLE instance that is considered the most appropriate neighbouring DLE node for DLE-a to joining the PDL service network.

4) DLAF sends notifications to the PDL service network to inform the arrival/join-in of DLE-a.

5) DLAF sends DLE-a a joining response with joining credential.

6) DLE-a sends a connecting request to DLE-b with its credential.

7) DLE-b sends a connecting response with the confirmation results of whether DLE-a is added as a neighbour of DLE-b and a peer node in the PDL service network.

8) DLE-a can start to participate the PDL service network with certain roles approved by DLAF.

8.3.2.2 A new DLE joining in via a peer DLE

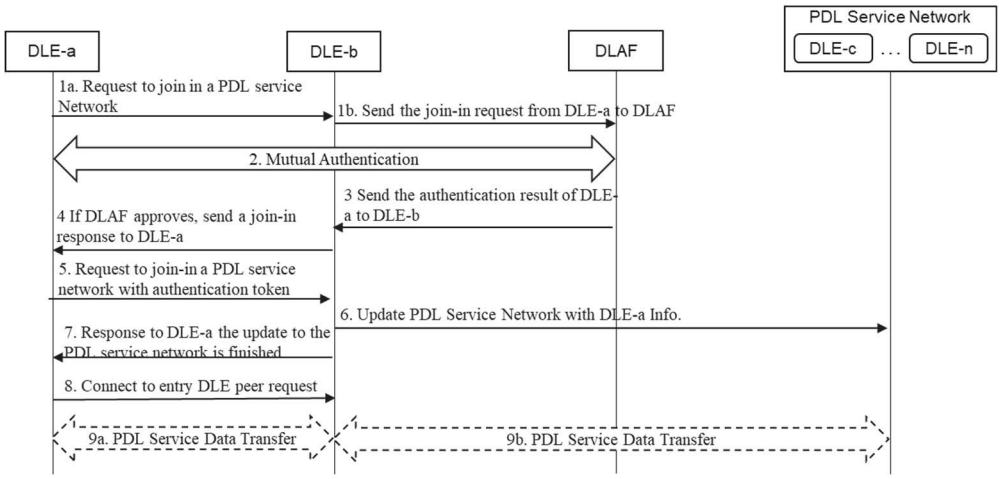


Figure 20: Procedure for a DLE joining a PDL service network via a peer DLE

The procedure illustrated in Figure 20 contains the following steps:

1) DLE-a sends a joining request to a neighbouring DLE-b. The neighbouring DLE-b is a gateway DLE configured to receive the joining request from other DLEs; and b) DLE-b further sends the joining request to DLAF.

2) DLAF and DLE-a proceed mutual authentication. Before the actual authentication, there are different options how DLE-a and DLAF can establish a communication channel:

   a) DLE-b can inform DLE-a the address of DLAF so that DLE-a can know where the authentication request can be sent.

   b) DLE-b can play as an intermediate node where authentication messages can be relayed by DLE-b (The following steps use this option).

3) DLAF sends its authentication result to DLE-a via DLE-b.

4) DLE-b sends the authentication results of DLAF to DLE-b if DLAF approves its joining request.

5) DLE-a sends a joining request to DLE-b with a token issued by DLAF.

6) DLE-b verifies the token provided by DLE-a and informs the PDL service network the arrival of DLE-a.

7) DLE-b sends a response to DLE-a to accept its joining request if the verification succeeds in Step 6.

8) DLE-a sends a connection request to DLE-b to become the peer node of DLE-b in the PDL service network.

9) DLE-a can start to participate the PDL service network with certain roles approved by DLAF.

8.3.3 PDL Service Network Scale-Down

8.3.3.1 Direct DLE leaving a PDL service network

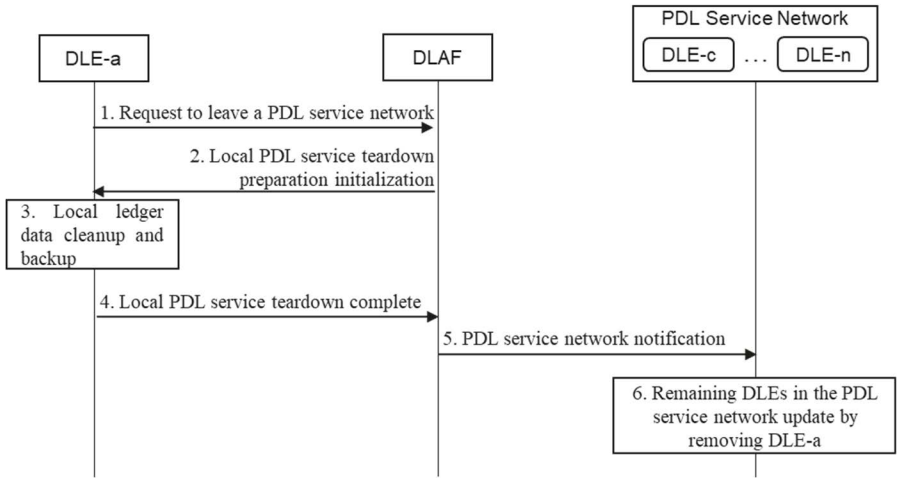


Figure 21: Procedure for a DLE leaving a PDL service network via DLAF

1) DLE-a sends a leaving request to DLAF for leaving a PDL service network. DLE-a informs DLAF that it is about to leave the PDL service network. In the leaving request, DLE-a has to provide its service status such as participated PDL service ID and the ledger status such as how much ledger data it stores and what the latest status of the ledger data and so on.

2) DLAF sends a response to DLE-a for PDL service teardown preparation. According to the provided service status and ledger status, DLAF provides a teardown instruction for DLE-a and sends in a response to DLE-a. The instruction will clarify what tasks need to be done before DLE-a quits the PDL service network. For example, the tasks can contain the location where the ledger data should be offloaded, the way to handle incoming transactions and so on.

3) DLE-a clears up the local ledger data. Given the received teardown instruction from DLAF, DLE-a one by one executes the task items in the instruction and finishes the teardown jobs locally.

4) DLE-a sends a PDL service teardown response. Once the teardown instruction is fully executed, DLE-a sends a response to DLAF to inform that the teardown preparation is ready and DLE-a is about to leave the PDL service network.

5) DLAF sends a notification to the PDL service network. After DLAF receives the confirmation from DLE-a that the teardown preparation is done, DLAF sends a notification the whole PDL service network where the notification informs the DLE instances with the information of the leaving a DLE peer (i.e. DLE-a). This notification can be a broadcast message or a message only sent to several hub nodes who can further propagate this message to the rest of the DLE instances.

6) Other DLE instances remaining in the PDL service network updates with the notification from DLAF. This can be that other DLE instances remove DLE-a from their peer list and so on.

8.3.3.2 Indirect DLE Leaving a PDL Service network

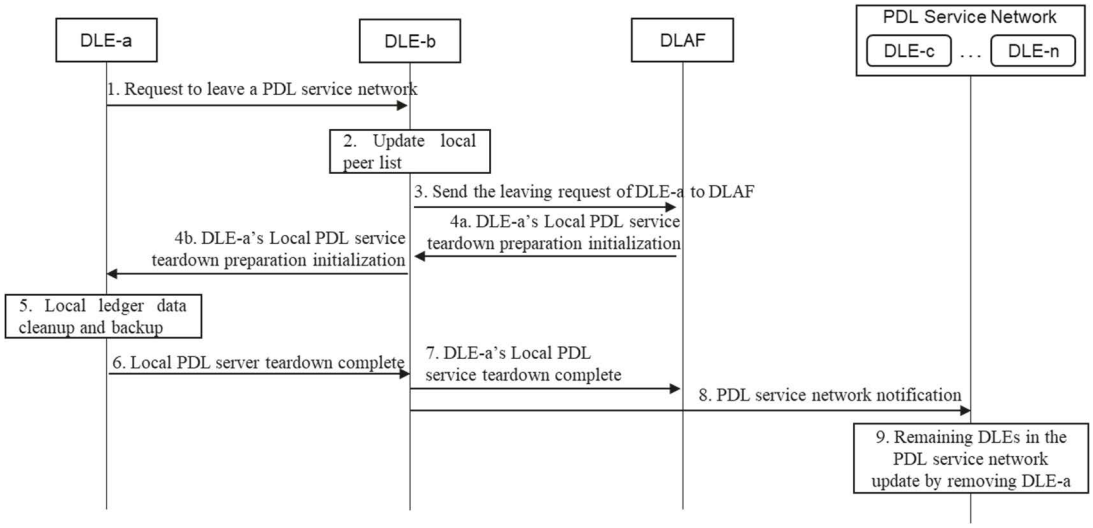


Figure 22: Procedure for a DLE leaving a PDL service network via a peer DLE

1) DLE-a sends a leaving request to a peer DLE-b for leaving a PDL service network. DLE-a informs DLE-b that it is about to leave the PDL service network. In the leaving request, DLE-a has to provide its service status such as participated PDL service ID and the ledger status such as how much ledger data it stores and what the latest status of the ledger data and so on.

2) DLE-b updates its local DLE-peer list. DLE-b removes DLE-a from its local peer list of the corresponding PDL service network. Meanwhile, DLE-b ignores any messages of the PDL service involving DLE-a.

3) DLE-b forwards the leaving request from DLE-a to DLAF with the parameters provided by DLE-a.

4) According to the provided service status and ledger status:

   a) DLAF provides a teardown instruction for DLE-a and sends in a response to DLE-b. The instruction will clarify what tasks need to be done before DLE-a quits the PDL service network. For example, the tasks can contain the location where the ledger data should be offloaded, the way to handle incoming transactions and so on.

   b) DLE-b forwards the response from DLAF to DLE-a with the teardown instruction provided by DLAF.

5) DLE-a clears up the local ledger data. Given the received teardown instruction from DLAF, DLE-a one by one executes the task items in the instruction and finishes the teardown jobs locally.

6) DLE-a sends a PDL service teardown response to DLE-b. Once the teardown instruction is fully executed, DLE-a sends a response to DLE-b to inform that the teardown preparation is ready and DLE-a is about to leave the PDL service network.

7) DLE-b forwards the teardown complete response to DLAF.

8) DLE-b sends a notification to the PDL service network. After DLAF receives the confirmation from DLE-a that the teardown preparation is done, DLE-b sends a notification the whole PDL service network where the notification informs the DLE instances with the information of the leaving a DLE peer (i.e. DLE-a). This notification can be a broadcast message or a message only sent to several hub nodes who can further propagate this message to the rest of the DLE instances.

9) Other DLE instances remaining in the PDL service network updates with the notification from DLE-b. This can be that other DLE instances remove DLE-a from their peer list and so on.

8.4 PDL service address management procedure

After a DLEC (e.g. UE) is onboarded to a native distributed ledger network, the DLEC has an authorized distributed ledger address, through which the DLEC can access and interact with the native distributed ledger (e.g. send transactions to distributed ledgers to store immutable data). In the meantime, the DLEC as a UE has regular 3GPP UE identifiers (e.g. SUPI/SUCI, 5G-GUTI) used between UE and the core network, or between core network functions; also, the DLEC may have multiple distributed ledger addresses (e.g. one for each native wireless distributed ledger network) used between the DLEC and the native distributed ledger, or between distributed ledger-related functions. UE identifier and distributed ledger address shall be associated or mapped with each other, so that given a UE identifier, one or multiple distributed ledger addresses associated with it should be easily found or vice versa.

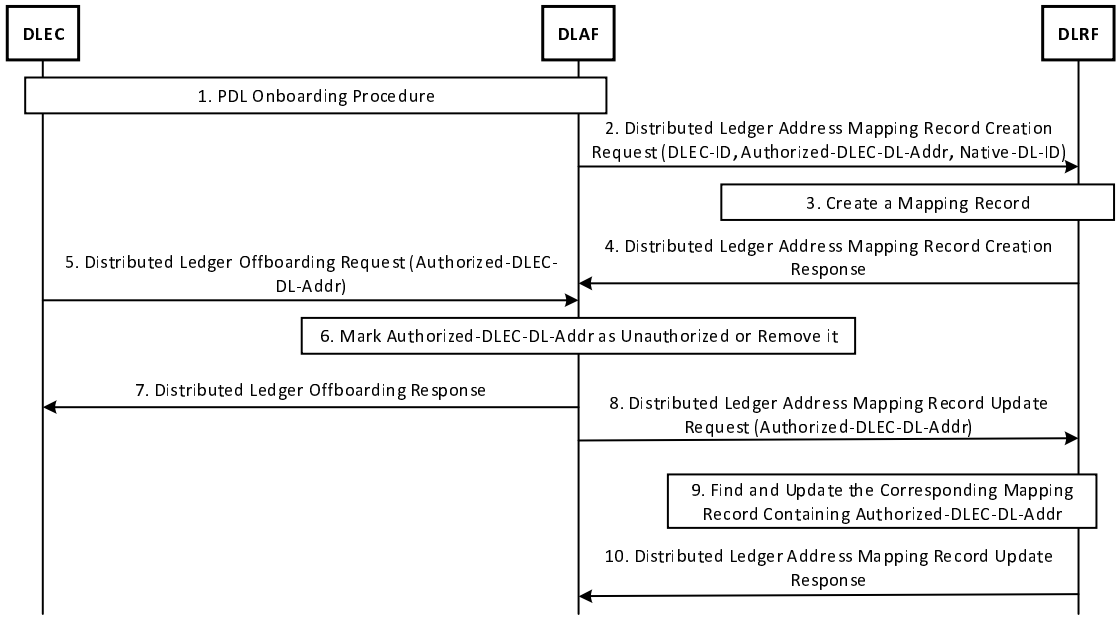


Figure 23: Create Mapping Record between Blockchain Address and 3GPP Identifier

Figure 23 illustrates a procedure for creating mapping record between distributed ledger address and 3GPP identifier. This procedure consists of the following steps.

1) DLEC completes a successful PDL onboarding with DLAF using the PDL service onboarding procedure in clause 8.1.4. As a result, DLEC's distributed ledger address has been authorized, referred to as Authorized-DLEC-DL-Addr.

2) DLAF sends a distributed ledger address mapping record creation request to DLRF. This request shall contain the following information:

   a) DLEC-ID: The 3GPP identifier of DLEC as indicated during PDL onboarding process in step 1.

   b) Authorized-DLEC-DL-Addr: DLEC's distributed ledger address as authorized during PDL onboarding process in step 1.

   c) Native-DL-ID: The identifier of the name of native distributed ledger that DLEC has been onboarded to in step 1.

3) DLRF creates a new mapping record for DLEC. This new record shall contain: DLEC-ID, Authorized-DLEC-DL-Addr, and Native-DL-ID.

4) DLRF sends a distributed ledger address mapping record creation response to DLAF.

5) When DLEC needs to offboard itself from a native distributed ledger, DLEC sends a distributed ledger offboarding request to DLAF. This request shall contain Authorized-DLEC-DL-Addr.

6) DLAF marks Authorized-DLEC-DL-Addr as ""unauthorized"" and removed it from its local storage.

7) DLAF sends a distributed ledger offloading response to DLEC.

8) DLAF sends a distributed ledger address mapping record update request to DLRF. This request contains Authorized-DLEC-DL-Addr.

9) DLRF looks up its locally maintained mapping records to find the one containing or corresponding to Authorized-DLEC-DL-Addr. Then, DLRF removes this found mapping record.

10) DLRF sends a blockchain address mapping record update response to DLAF indicating the successful update of corresponding mapping record in step 9.

9 Integration recommendation of PDL capability with telecom networks

9.1 General information

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

Generally, given the assumed telecom network architecture, several new NFs will be added. The first one is a management and control function, which is called Distributed Ledger Anchor Function (DLAF). Another function is a Distributed Ledger Repository Function (DLRF), which is used to provide necessary software to a resource node. The third function is a Distributed Ledger Data Storage Management (DLDSM), which provides storage of PDL data to telecom network. Another new function introduced into telecom network is Distributed Ledger Enabler (DLE), which is the main function to realize a PDL service with the resource within the telecom network. DLAF controls DLE to provision PDL services in the telecom network infrastructure. As a nationwide infrastructure, telecom networks already become the 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 distributed ledger network consisting of a set of distributed but interconnected nodes. Hence, PDL service provisioning within telecom network is a non-trivial task, because an operator needs consider how a blockchain network can be mapped onto the resource pool, 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 distributed ledger capability as part of the native/fundamental features of the telecom network.

9.2 Telecom-native PDL capability

In the telecom network architecture, DLAF can be a NF in the core network. DLAF can manage and control DLE instances in the network to operate PDL services. DLRF and DLDSM are also deployed in core network, providing software storage and data storage functions for PDL service. DLE 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 PDL all-in-one node, and nodes such as UE, base station, NF and other nodes can only be used as users of the chain.

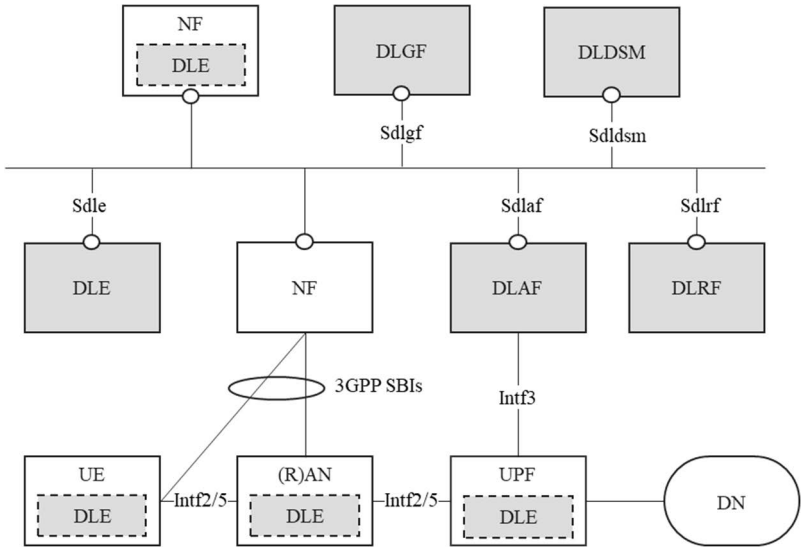


Figure 24: A Telecom-native PDL capability integration

As shown in Figure 24, DLAF, DLRF, DLDSM and DLE can be standalone NFs providing services via SBIs. For DLE, it has two options. The first option is that a DLE can be a standalone NF in the core network, acting as a full node in a PDL service network. The second option is that a DLE can collocate with an existing NF in the core network (e.g. an AMF / UPF), UE and/or RAN. In the latter case, the role of the DLE depends on the specific requirement of a provisioned PDL service.

9.3 Telecom-connected PDL capability

The PDL capability can also exist relatively independent to the existing core network architecture. In other words, the proposed architecture for PDL service provisioning can be connected via a single interface, for example, DLAF. In this case, a PDL service provisioning request will be simply transfer to DLAF and the existing core network will not be aware of the details of request. Instead, only a signalling label is required to divert the traffic to DLAF.

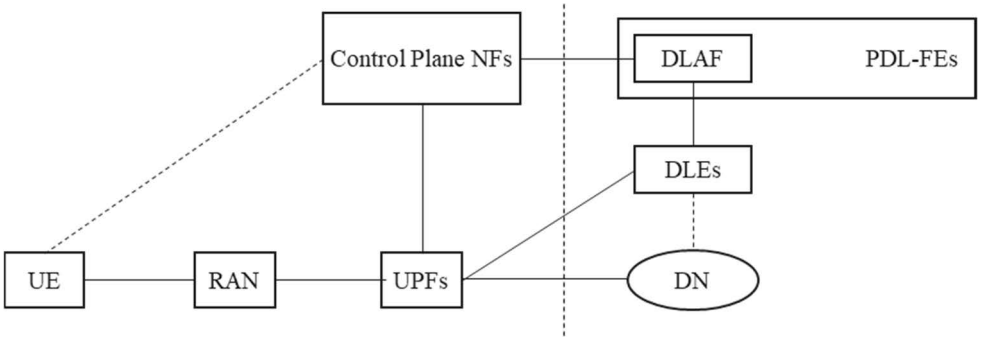


Figure 25: Telecom network-connected PDL capability integration

As shown in Figure 25, the PDL service architecture is standalone to a telecom network. The PDL service is deployed outside of the telecom network, where interactions between the two domains are done over interfaces between control planes and user plane. The interface between the control planes is used to coordinate users to access the PDL service; after the coordination is done, the user accesses the PDL service via data plane of the telecom network to the PDL service network either directly to DLEs or indirectly through data network.

9.4 Deployment Considerations of PDL 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 centre 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 DLRF Deployment Options

Logically, DLRF is an NF in the core network. Its instantiation can be either centralized or distributed.

A centralized option means that DLRF instances all locate at a centre 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, DLRF instances in this deployment option are far from the edge of the telecom network.

A distributed option means that DLRF instances distribute at different domains in the telecom network logically and/or geographically. Each DLRF may provide software libraries to the local PDL nodes. In practice, DLRF instances in this deployment option can be deployed closer to the edge of the telecom network.

In some cases, if there is no real-time need to install blockchain software libraries on the network, or if security considerations require that the DLRF not be exposed to other NFs in the core network, the DLRF can be deployed in the form of an independently controlled server managed by the operator, and only provide interfaces to the operator's management system.

9.4.3 DLDSM Deployment Options

Logically, DLDSM is an NF in the core network. Its instantiation can be either centralized or distributed.

A centralized option means that DLDSM instances all locate at a centre 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, DLDSM instances in this deployment option are far from the edge of the telecom network.

A distributed option means that DLDSM instances distribute at different domains in the telecom network logically and/or geographically. Each DLDSM may provide data storage service to the local PDL nodes. In practice, DLDSM instances in this deployment option can be deployed closer to the edge of the telecom network.

9.4.4 DLE Deployment Options

The placement of a DLE can locate at any type of nodes in the telecom network. For example, it can run on a UE, a base station, an NF either in control plane, user plane or both. A DLE can also be instantiated standalone as an individual function/network function, or even a server machine when natively co-locating with other functions does not meet the provisioning requirements. In any instantiating form that a DLE can be deployed, the execution mode of the DLE can be one of the modes specified in clause 6.2.1.

9.4.5 DLGF Deployment Options

DLGF can be located in difference places in the telecom network. For example, a DLGF can reside in the core network of a 3GPP network; a DLGF also can be placed in an edge network in the telecom network. A DLGF can be instantiated as a standalone function or network function. A DLGF can also be instantiated and integrated with other functions; for example, a DLGF and a DLAF can be combined; in another example, a DLGF can be implemented as a part of 3GPP Authentication Service Function (AUSF).

9.5 Mapping and Classification of PDL Functions in Telecom Networks

9.5.1 Introduction

Similar to the classification of existing 3GPP NFs of a telecom network, the proposed PDL function functions can also be classified into three categories: management, control and data/user planes. PDL functions play different roles at the three classes of planes for PDL service operation.

9.5.2 PDL function Classification

For Management: Processes related to PDL capabilities fall under management processes, including the installation, updating, and removal of blockchain packages, as well as configuring the blockchain capabilities of nodes to active, locked, or closed states.

For Control: Processes related to the PDL service itself fall under control processes, including the creation, deletion, and updating of a PDL service, the registration, deregistration, joining, or exiting of DLE nodes from PDL, as well as the subscription and notification of PDL-related information.

For User Data: The transmission of PDL data between DLEs falls under data processes, including the uploading of transactions and blocks to the ledger by DLEs.

9.5.3 Possible Mapping to Existing Operation Planes in Telecom Networks

According to the roles of the functional entities proposed in the PDL service provisioning architecture, they can be possibly mapped to the existing planes in the telecom networks as follows:

- First of all, PDL functions for management purposes can be mapped to either the management plane in telecom networks. At this time, when a PDL service needs to configure the software libraries of DLE instances of a PDL service network, a PDL service needs to update or remove software packages for DLE instances of a PDL service, or the configurations of a PDL service network such as the topology of DLE instances, these types of works are management tasks. In the proposed architecture of PDL service provisioning, several PDL functions e.g. DLAF, DLRF and/or DLGF can involve these tasks mentioned above.

- Second, in some situations, a PDL function for management purposes can also be mapped to the control plane in telecom networks. For example, DLAF directly issues configuration parameters to telecom network nodes (such as base stations, NFs, etc.), dynamically, in real-time, and automatically adapting a PDL service. This can be the situation where a client requests to connect to the PDL service and DLAF is responsible for interacting with SMF to establish a connection for the client to the PDL service network. In addition, DLRF can be deployed in the core network as an independent NF, providing blockchain software packages to DLAF or even directly to other telecom network nodes. Therefore, there is no clear boundary between a management task and an operational control task.

- Last but not least, DLE in the PDL service provisioning architecture belongs to the data/user plane in telecom networks. This because DLE is the main PDL function where the actual PDL service is realized with specific protocols, transaction data generated for the PDL service rather than the operator. A difference is that a DLE can exist on a UE either as a DLE-client or DLE-peer, where the latter can provide the PDL service to other users around.

The general observation is that due to the special characteristics of a PDL service, its operation involves a mixture of management and control plane tasks. Therefore, PDL functions of the propose PDL service provisioning architecture can play roles at different planes at the same time.

9.6 PDL service deployment considerations

9.6.1 Single-operator provisioning

In this case, a PDL service is provisioned only with the resources from one operator's domain. A PDL service provisioning request is handled by the DLAF in the core network, and according to the requirements of the PDL service, DLAF schedules the underlying resource layer and launch PDL service network with multiple DLE instances as shown in Figure 24.

Note that in this scenario, the PDL service can be used for internal purposes such as sub-networks in different areas sharing information with distributed consensus protocols where the accountability and immutability features are needed even for the operator its own. Another use case could be that a 3rd-party user trusts the network infrastructure (i.e. the national telecom operator); the 3rd-party user simply deploys a PDL service with the network infrastructure to run its own PDL service for its own applications.

9.6.2 Multi-operator/party provisioning

The first sub-case is that multiple operators establish PDL services through offline negotiations, or DLAFs of multiple operators negotiate through telecommunications network gateways (such as SEPP) to determine the functionalities required for PDL services (consensus, security algorithms, ledger technology, etc.) as well as the required DLEs to be invoked. The selected DLEs also transmit blockchain data through the gateway.

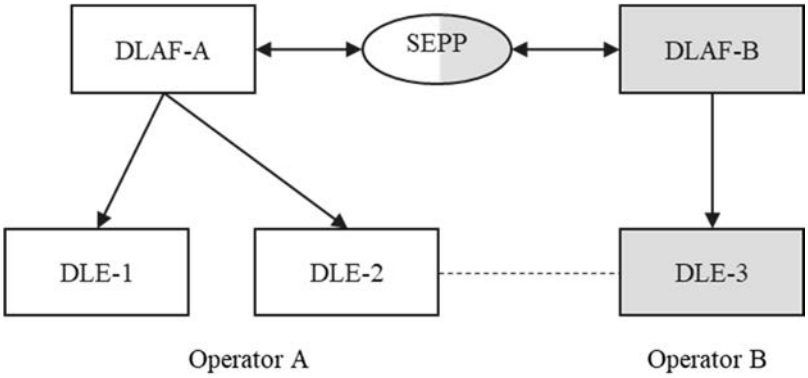


Figure 26: Multi-Operator PDL service provisioning organization

The second sub-case is that operators and 3rd-parties establish PDL services through offline negotiations, or the 3rd-party server and the operator's DLAF negotiate through DN to determine the functionalities required for a PDL service (consensus, security algorithms, ledger technology, etc.) as well as the required DLEs to be invoked. The selected DLEs transmit blockchain data to the third party's blockchain node through DN.

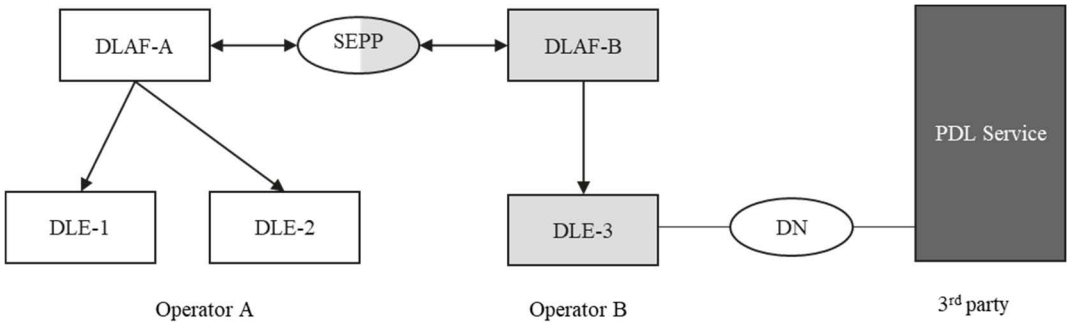


Figure 27: Multi-Party PDL Service Provisioning Organization

9.7 Summary

The recommendation in this clause suggests several ways where the proposed PDL functions could be integrated with in the existing telecom network architecture so that a telecom network is enhanced with PDL capability to provide PDL services.

10 Conclusion

10.1 General information

The present document first proposes the reference models of PDL service provisioning where typical scenarios are covered. After that, several important high-level features of the proposed architecture are described. Given the proposed architecture, PDL functions are detailed with their provided services (e.g. interfaces). Working procedures are defined based on the services of the PDL functions. At the end, several possible ways to integrate the proposed PDL service provisioning architecture with the existing telecom network architecture are discussed, which can be further utilized as a reference when 3GPP works on the standardization of enhancing 3GPP system with PDL capability.

10.2 Recommendation for the next steps

Clause 9 specifies several ways to integrate the PDL capability into telecom network architecture based on the proposed PDL service provisioning architecture. It is recommended to further study how the proposed PDL service provisioning architecture works together with the telecom network architecture to provide PDL-enhanced telecom network services. Specifically, the following technical aspects shall be considered for standardizations by ETSI ISG PDL and/or other standardization bodies to whom it sees relevant:

1) Specification of signalling protocol design for PDL service procedures in telecom network.

2) Specification of integrating PDL service procedures with telecom network service procedures.

3) Specification of integrating PDL service signalling with telecom network service protocols.

History

Document history		
V1.1.1	November 2024	Publication