ETSI GS PDL 013 V1.1.1 (2022-10)

GROUP SPECIFICATION

Permissioned Distributed Ledger (PDL); Supporting Distributed Data Management

Disclaimer

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

Reference

DGS/PDL-0013_Sup_Dis_Data_Mgmt

Keywords

data management, PDL

ETSI

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

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

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

Important notice

The present document can be downloaded from:
http://www.etsi.org/standards-search

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

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

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

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

Notice of disclaimer & limitation of liability

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.
No recommendation as to products and services or vendors is made or should be implied.
No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.
In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

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

Copyright Notification

Contents

Intellectual Property Rights

Essential patents

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

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

Trademarks

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

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

Foreword

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

Modal verbs terminology

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

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

Executive summary

The present document defines requirements and functional architecture of supporting distributed data management based on Permissioned Distributed Ledger (PDL) reference architecture. This includes expanded ETSI ISG-PDL platform services for supporting distributed data management.

Introduction

The present document specifies PDL-based distributed data management. The organization of the present document is as follows. Clause 1 defines the scope of the present document. Clauses 2 and 3 provide normative and informative references and definition of terms, respectively. Clause 4 provides an overview of PDL reference architecture. Clause 5 describes distributed data management use cases and requirements. Clause 6 lists architectural requirements of PDL-based distributed data management. Clause 7 defines expanded ETSI ISG-PDL platform services for PDL-based distributed data management.

1 Scope

The present document specifies distributed data management based on PDL reference architecture. This includes:

- defining architectural requirements that are derived from distributed data management use cases including related use cases such as those described in ETSI GR PDL 009 [i.1] and ETSI GR PDL 002 [i.2];
- defining PDL-based distributed data management architecture according to PDL reference architecture as defined in ETSI GS PDL 012 [1]); and
- defining expanded ETSI ISG-PDL platform services for PDL-based distributed data management.

2 References

2.1 Normative references

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

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

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

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

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

[2] ETSI GS PDL 011: "Permissioned Distributed Ledger (PDL); Specification of Requirements for Smart Contracts' architecture and security".

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] ETSI GR PDL 009 (V1.1.1): "Permissioned Distributed Ledger (PDL); Federated Data Management".

[i.2] ETSI GR PDL 002 (V1.1.1): "Permissioned Distributed Ledger (PDL); Applicability and compliance to data processing requirements".

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:

API	Application Programming Interface
ARPS	Application Registration Platform Service
DC	Data Collector
DCC	Data Computation Controller
DCN	Data Computation Node
DCS	Data Consumer
DD	Data Discoverers
DDAPP	Distributed Data Application
DDM	Distributed Data Management
DH	Data Host
DLT	Distributed Ledger Technology
DO	Data Owner
DP	Data Provider
DPS	Discovery Platform Service
DS	Data Source
ETSI	European Telecommunications Standards Institute
ETSI ISG-PDL	ETSI Industry Specification Group for Permissioned Distributed Ledger
FCAPS	Fault, Configuration, Accounting, Performance and Security
FL	Federated Learning
GDPR	General Data Protection Regulation
GR	Group Report
GS	Group Specification
IRP	Interface Reference Point
ISG	Industry Specification Group
MARL	Multi-Agent Reinforcement Learning
MPS	Messaging Platform Service
PDL	Permissioned Distributed Ledger
RD	Requirement on Decentralization
RDCS	Requirement on Data Control and Sovereignty
RDDA	Requirement on Distributed Data Application
RDI	Requirement on Data Integrity
RDMA	Requirement on Data Management Automation
RDP	Requirement on Data Privacy
RDPV	Requirement on Data Provenance
RI	Requirement on Incentivization
RL	Reinforcement Learning
RPS	Registration Platform Service
RPSL	Requirement on Platform Service Layer
RT	Requirement on Trust
RUDLTN	Requirement on Underlying DLT Networks
SPS	Storage Platform Service
TCI	Transaction Creation Indication
TMPS	Transaction Management Platform Service

4 PDL Reference Architecture

4.1 Introduction

ETSI GS PDL 012 [1] develops a layered PDL reference architecture, which consists of five layers as illustrated in Figure 4.1-1. Each layer is designed in a manner that allows abstraction, such that it can be operated regardless of the implementation specifics of the other layers. In addition, Interface Reference Points (IRPs) are defined between different layers:

- **PDL Applications:** Various PDL-based applications leverage PDL services as provided by the below described Service Layer to interact with different DLT networks. For example, a PDL-based data sharing application utilizes a DLT network as a distributed infrastructure to enable distributed sharing. An application may also interact with external storage to store certain data that requires better privacy control or to reduce the overhead to DLT networks.

- **Application Abstract Layer:** This layer utilizes Data Model Brokers/Gateways enabling applications that allow different data models to communicate with ETSI ISG-PDL compliant platforms. This layer is located between the PDL Applications and Platform Services Layers and implemented through the Data Model Broker Platform Service where necessary.

- **Platform Services Layer,** which provides useful services for applications to support various types of applications using PDL technology. As a result, an application could leverage services from the Platform Service Layer rather than embed such services within the application itself. This reduces applications' complexity, accelerates application development and deployment, and increases interoperability. For example, the Platform Service Layer may include a Transaction Management Platform Service to facilitate transaction creation in a manner transparent to a specific PDL type (i.e. a specific deployed DLT network) and in a manner uniform across applications using such platform; this is an example of layer abstraction in its essence. Such a Transaction Management Platform Service can perform transaction transformation/adaptation between applications running on different PDL types to facilitate application operations in a complex environment.

- **DLT Abstraction Layer,** which consists of a Data Model Broker/Gateway enabling Platform services to communicate with ETSI ISG-PDL compliant PDL types regardless of the specific type of the underlying PDL. An additional functionality of such abstraction layer is to allow interoperability between different DLT types, which may differ not only in data model structure but also on consensus mechanism and smart-contract functionality. Such abstraction layer hides the differences between PDL types and provides a unified service-facing interface on the services side and a PDL specific interface on the PDL side. This layer is located between the "Techno" and the "Disco" IRPs and implemented through the Data-Model Broker Platform Service where applicable.

- **DLT Layer,** which includes various DLT networks (e.g. an implementation of a specific DLT type) and potentially the abstraction of DLT networks. While DLT networks and chains may vary in terms of consensus mechanism and smart contract format, the abstract functionality of a chain is very similar across most DLTs: Storing a distributed chain of data blocks in a tamper-resistant manner, and performing pre-programmed actions based on rules (i.e. "Smart Contracts") on all copies of the distributed chain. Yet, not all DLT types are necessarily compliant with the ETSI ISG-PDL layered architecture approach, thus the DLT layer can only include and accommodate DLT types that are compliant with said architecture.

- **Interface Reference Points (IRPs),** which define communication channels through which the functional blocks defined above communicate with each other. The IRPs are given names for reference purposes (e.g. Debka, Tango, etc.).

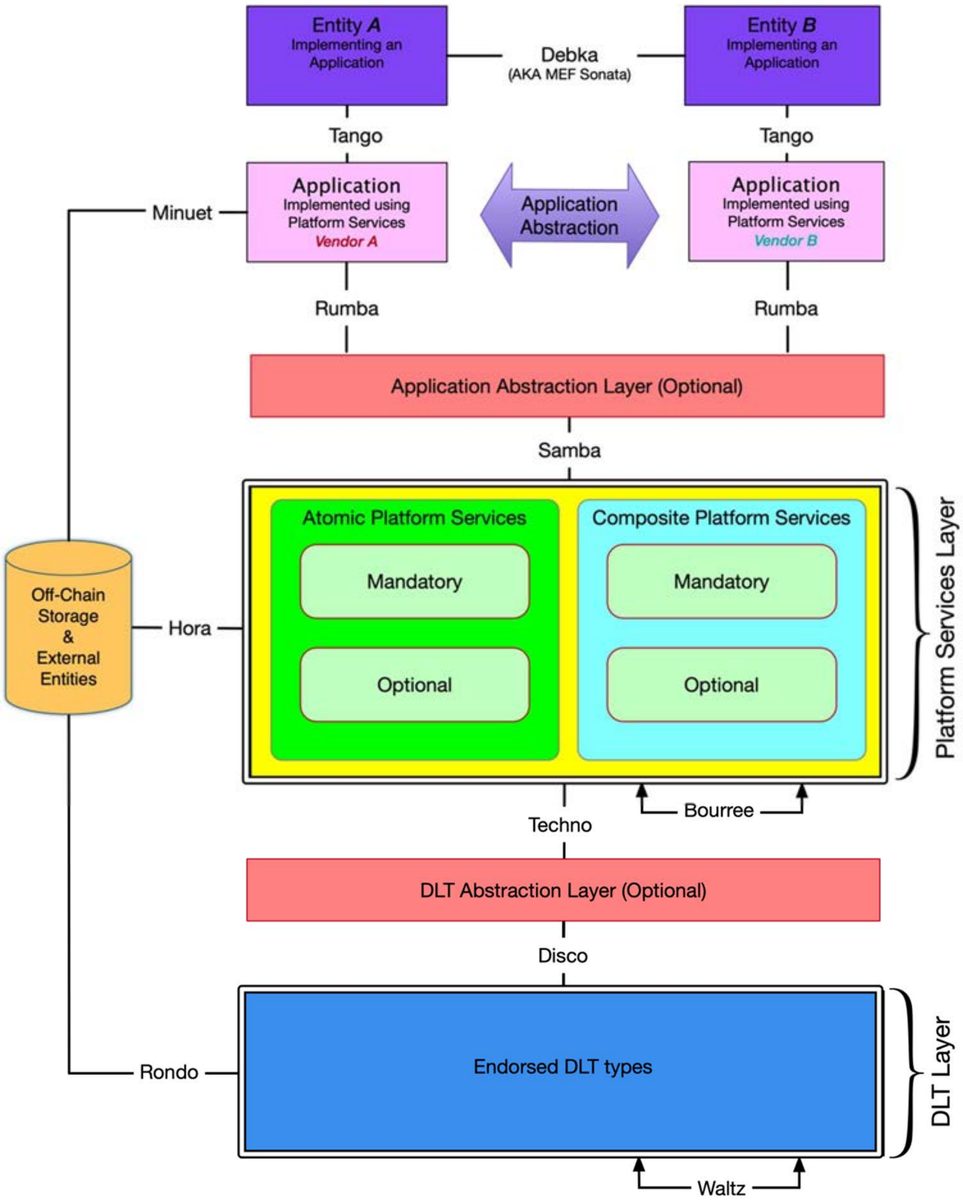

Figure 4.1-1: ETSI ISG-PDL Reference Architecture (Source: ETSI GS PDL 012 [1])

Four ETSI ISG-PDL platform categories (namely: Alpha, Bravo, Charlie, and Delta) are defined in ETSI GS PDL 012 [1]. The differences among those four categories lie in a few factors:

1) the number of involved vendors;

2) the number of supported underlying DLT technologies; and

3) the number of supported applications.

- Alpha Platforms: that are designed, developed, delivered, and integrated to all users of the said platform by a single vendor using a single DLT technology.

- Bravo Platforms: that are designed, developed, delivered, and integrated to all users of the said platform by a single vendor, but can operate using two or more underlying DLT technologies.

- Charlie Platforms: that can operate using two or more underlying DLT technologies and are designed and developed towards a specification of an application abstraction layer so that any application that supports such an abstraction layer can interface with the said platform. Moreover - the Platform Services in a Charlie Platform can be developed by multiple vendors towards the specification defined in ETSI GS PDL 012 [1].

- Delta Platforms: That use a single DLT technology and are designed and developed towards a specification of an application abstraction layer so that any Application that supports such an abstraction layer can interface with the said platform. This is, in essence, a simplified Charlie Platform that uses a single DLT type thus eliminating the DLT abstraction layer and eliminating the overheads associated with DLT interoperability.

4.2 Platform Services Layer

The Platform Services Layer hosts several types of services; details of each service are described in ETSI GS PDL 012 [1] where each such Platform Service is defined:

- The PDL Platform Services can be Atomic services or Composite services. Each such service could be Mandatory or Optional. Atomic services are self-sufficient and do not rely on other Platform services for their proper operation, while Composite services use one or more other Platform service to operate. A platform cannot function properly unless all Mandatory Platform Services are implemented therein, while Optional services may only be required for specific purposes or use-cases.

- PDL Platform Services are services and functionalities provided by the PDL platform that all applications may use. Platform Services may reuse or be built upon other Platform Services. Examples of Platform Services include: namespace, identity, location, discovery, messaging, policy, governance, security, composition, access control, concurrency storage, modelling, distributed processing, resource management, service management, transaction management, etc.

- In addition, PDL Platform Services Layer has Application Specific Platform Services that are services used by specific applications or specific groups of applications and are not needed or cannot be made useful for other applications (e.g. measurement of precipitation is useful for agriculture and weather applications but has no use for data storage applications). Such services may be implemented within the application itself, however the developer may want to contribute them and install them on the platform so the can be re-used by other applications in the future if the need arises.

Table 4.2-1 lists the Platform Services as defined in ETSI GS PDL 012 [1].

Table 4.2-1: ETSI ISG-PDL Platform Services [1]

PDL Platform Service name	Mandatory (M) or Optional (O)	Atomic (A) or Composite (C)	Short description
Namespace	M	A	Ensures that all of a given set of objects for a particular function have unique names.
Identity	M	A	Unambiguously identifies an instance of an entity from all other instances of this and other objects.
Location	O	A	Associates an object with a location.
Registration	O	A	List a managed object with authorities or registries.
Discovery	O	A	Discovery of services offered by the services layer and discovery of PDL networks.
Messaging	M	C	Enables communication between a group of entities.
Policy	O	C	Manage and control the changing and/or maintaining of the state of managed objects.
Security	M	C	A collection of services that assess, reduce, protect, and manage security risks.
Authentication	M	C	Verifies that a subject requesting to perform an operation on a target is who they say they are.
Authorization	O	C	Permitting or denying access to a target by a subject.
Cryptography	O	C	Managing protocols that prevent third parties from reading private communications.
Encryption	O	C	Encoding information using a key into an unintelligible form.
Identity Management	O	C	Access control based on the identity of an entity.
Key Management	O	C	Management of cryptographic keys.
Logging	O	C	Dynamic ingestion and collection of logs.
Governance	M	C	Rules and tools that control the behaviour and function of a PDL.
Implementation Agreements	O	C	Rules and agreements that describe how ETSI ISG-PDL Services are implemented and control the behaviour of a PDL platform.

PDL Platform Service name	Mandatory (M) or Optional (O)	Atomic (A) or Composite (C)	Short description
Governing Entity	M	C	Defines the rules and implementation agreements. Ensures compliance. Resolves conflicts where needed.
Composition	O	C	Defines who can compose new services and how such new services are composed.
Access Control	M	C	Defines who can perform which operations on which set of *target* entities.
Fault Tolerance	O	C	Defines how to handle faulty instructions.
Distribution Transparency	O	C	defines how to maintain transparency when distributing information to target entities.
Publish and Subscribe	O	C	Defines how entities publish services and subscribe to services.
Concurrency	O	C	Defines how entities handle concurrency.
Storage	M	C	A group of services related to Storage.
In Memory Storage	M	C	Data that is stored in the random access memory of a computer running an application.
File System Storage	M	C	Storage on a directly connected storage device.
On-Chain Storage	M	C	Application data that is stored in blocks on all nodes using the chain.
Off-Chain storge	O	C	Information in a digital, machine-readable medium that is not stored on the main chain.
Distributed Blockchain Storage	M	C	Storage on a Distributed Blockchain ledger.
Modelling	M	C	A group of services related to Modelling.
Information Model	M	C	Presentation of concepts of interest to platform management environment in a *technology-neutral* form as objects and relationships between objects.
Data Model	M	C	Representation of applicable concepts in a *technology-specific concrete* form.
Model Search	O	C	Enables search for specific or generic models within existing information and data models.
Model Stitching	O	C	Enables integrating multiple models or parts of models into a single model.
Topology	M	C	Allows a node to identify other nodes on the PDL and identify which nodes to communicate with when performing PDL related tasks.
Event Processing	M	C	Processes node-specific and platform-wide events as they occur.
Distributed Data Collection	O	C	Performs tasks related to collection of data that are location-independent.
Distributed Secret Sharing	O	C	Sharing of confidential data between nodes in a manner that maintains confidentiality of the data.
Resource Management	M	C	Defines how to administer and manage Resources.
Resource Discovery	O	C	Enables discovery of resources available to applications and nodes.
Resource Virtualization	O	C	Creating a virtual resource that mimics the behaviour of a physical resource.
Resource Inventory Management	O	C	Management of node-specific and platform-wide resource inventory.
Resource Admin and Management	M	C	Administration and management of node-specific and platform-wide resources.
Resource FCAPS	O	C	Resource management tasks defined by the ISO model.
Resource Composition	O	C	Creation and management of composite resources.
Platform Services Management	M	C	Defines how to administer and manage Platform Services.
Platform Service Discovery	M	C	Provides means to discover services available to applications and nodes.
Platform Service Virtualization	O	C	Creating a service using virtual resources.
Platform Service Inventory Management	O	C	Keeping track of inventory and serviceability of Platform services.
Platform Service Admin and Management	M	C	Administration and management of Platform Services through governance.

PDL Platform Service name	Mandatory (M) or Optional (O)	Atomic (A) or Composite (C)	Short description
Platform Service FCAPS	O	C	Platform Service management tasks defined by the ISO model.
Platform Service Composition	O	C	Creation and management of the composition of Composite Platform Services.
Application Management	M	C	Creation and management of Applications.
Application Composition	M	C	Composing an Application from two or more managed objects.
Application and Service Orchestration	O	C	Orchestrating multiple managed objects so they provide a desired set of behaviours.
Orchestration	O	C	Orchestration of objects, resources, services, and/or applications so that they collectively provide the desired functionality and behaviour.
Platform Exploration	O	C	Allows an application to indicate its requirements and explore whether the platform offers such service capabilities
Application Registration	O	C	Registers and lists all applications operated on a platform.
Transaction Management	O	C	Facilitates transaction related interactions between applications/services and underlying PDL networks.
Data Model Gateway/Broker	O	C	Defines tools that enable two systems with different data models to interact.
API Presentation	O	C	A specific Data Model Gateway/Broker implementation for environments that use APIs to exchange data between objects.
Application Specific Services	O	C	Serve a specific application or a group of applications but not required or used by other applications using the platform.

5 Distributed Data Management

5.1 Introduction

Distributed Data Management (DDM) is referred to operation and manipulation of data in distributed manners such as those illustrated in Figure 5.1-1. For each scenario in Figure 5.1-1, there are multiple distributed parties (referred to as data nodes); each data node has a Distributed Data Application (DDAPP), which supports a specific distributed data management task among those distributed data nodes:

- Distributed Data Discovery: Data is discovered from multiple distributed parties. Data discovery is required for distributed data collection, distributed data storage, distributed data sharing, and distributed data computation.

- Distributed Data Collection: Data is collected from multiple distributed parties.

- Distributed Data Storage: Data is stored in multiple distributed parties.

- Distributed Data Sharing: Data is distributed and shared among multiple parties.

- Distributed Data Computation: Multiple parties perform data computation in a distributed and collaborative way, for example, federated learning, distributed machine learning, and multi-party computation.

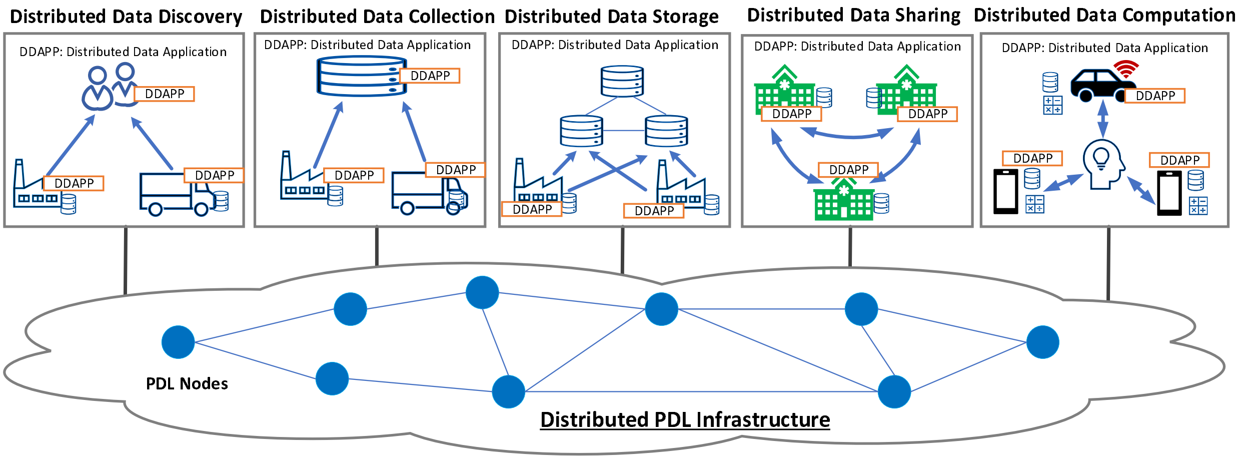
Distributed Data Discovery
Distributed Data Collection
Distributed Data Storage
Distributed Data Sharing
Distributed Data Computation
DDAPP: Distributed Data Application
DDAPP
PDL Nodes
Distributed PDL Infrastructure

Figure 5.1-1: Distributed Data Management

5.2 Distributed Data Discovery

Data discovery occurs before performing other data management operations such as data collection, data sharing, and data computation. For example, before original data is collected, it needs to be discovered. In addition, data discovery may also be needed for data storage such as moving a specific type of data from one place to another. Two types of data nodes are involved in distributed data discovery: Data Hosts (DHs) and Data Discoverers (DDs). An entity which hosts data to be discovered can be referred to as a DH. DDs discover expected data from DHs. Data discovery applications exist in both DDs and DHs for jointly performing distributed data discovery.

Figure 5.2-1 illustrates distributed data discovery, where one or more DDs discover data from multiple distributed DHs. For example, the data discoverer DD-B discovers its expected data from three DHs (i.e. DH-2, DH-3, and DH-4). Distributed DHs can register their data to one or more data repositories, from which DDs can discover expected data. When a DD knows the address of a DH, it can also discover data directly from such DH. In some cases the data required/expected by a DD is not fully available on a single DH and may require discovery of multiple DHs.

- If data discovery is used as a precursor for distributed data collection, DHs are Data Sources (DSs) while a DD can be a Data Collector (DC).

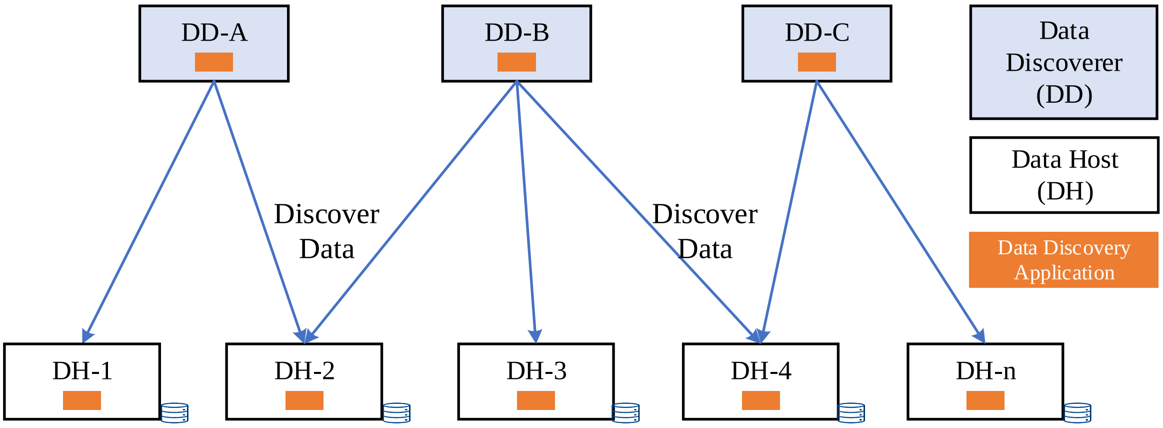

Figure 5.2-1: Distributed Data Discovery

5.3 Distributed Data Collection

Two types of data nodes are involved in distributed data collection: Data Collectors (DCs) and Data Sources (DSs). For the purpose of data collection, a DC is responsible for actively retrieving or passively receiving data from one or more DSs. In the passive receipt scenario a DS transmits its original data to one or multiple DCs. In the active retrieve scenario a DC has to retrieve the data from the DS. DCs may maintain collected data locally and may store or forward the collected data to other DCs or external data storage systems.

In a distributed data collection scenario, data is collected in a distributed manner, as illustrated in Figure 5.3-1. In such a scenario each DC may collect data from different DSs while the data collected by each DC is distributed to all DCs. Data collection applications exist in both DSs and DCs to jointly perform distributed data collection. The resulting data collected by all DCs from all DSs is then stored in a distributed manner on all the nodes that are used for storage. Note that some nodes may be used to both collect and store data, some may be used for data collection purposes only and may not be used for data storage while other nodes may be used for data storge only and will not participate in data collection:

- Distributed DCs: Data collection will be performed by multiple distributed DCs. Those DCs could be fully decentralized or form a hierarchical structure. For example, Figure 5.3-1 shows four distributed DCs (i.e. DC-A, DC-B, DC-C, and DC-D). Each DC maintains the collected data from a group of DSs. The collected data can be replicated and/or moved among multiple distributed DCs.

- Distributed DSs: Many DSs are distributed (e.g. due to geographical spread or due to operational circumstances). A set of DSs (e.g. DS-A1, DS-A2, and DS-An) can be logically grouped together and transmit their original data to a DC (e.g. DC-A).

- Distributed Data Transmission: The original data is transmitted from DSs to DCs in a distributed way. There could be three different transmission modes:

  1) Each DS (e.g. DS-A1) transmits its original data directly to a DC (e.g. DC-A).

  2) One DS (e.g. DS-Bn) can send its data to another DS (e.g. DS-B2), which then forwards the data to a DC (e.g. DC-B).

  3) One DS (e.g. DS-Cn) can send its original data to multiple DCs (e.g. DC-B and DC-C) for the purpose of either load balancing or diversity.

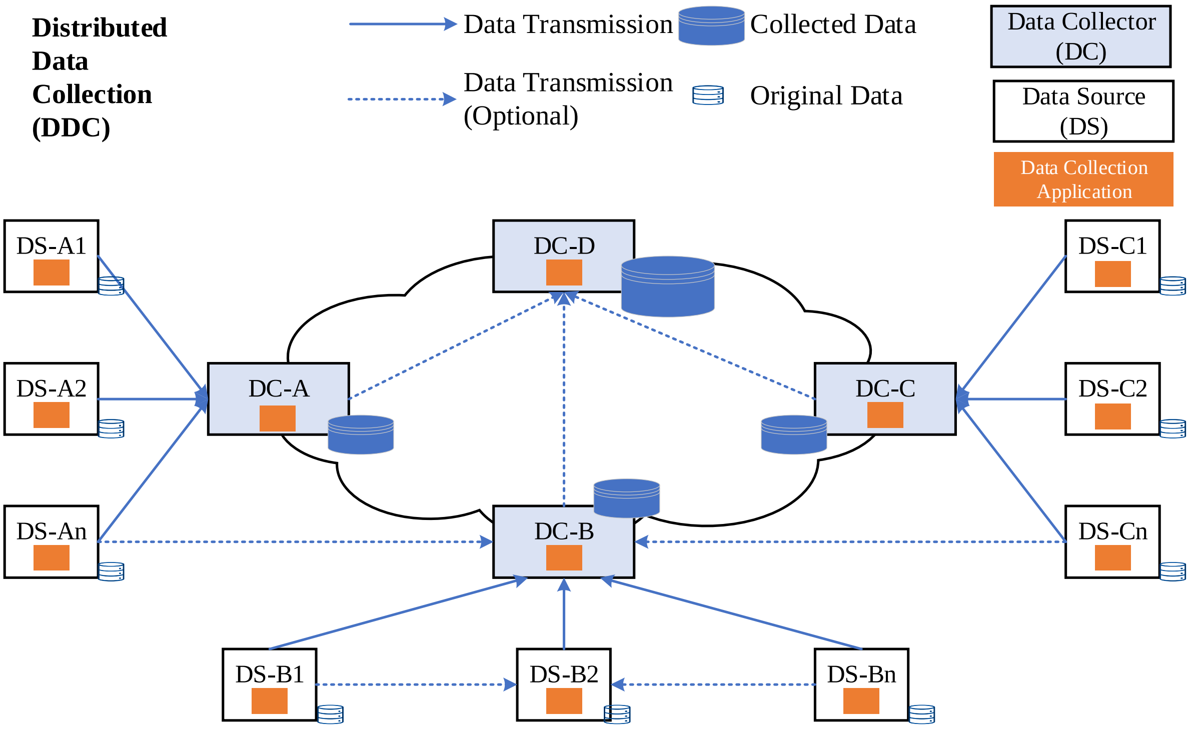

Figure 5.3-1: Distributed Data Collection

5.4 Distributed Data Storage

Two types of data nodes are involved in distributed data storage: Data Providers (DPs) and Data Hosts (DHs). A DP generates the data to be stored. DPs transmit their data for storage on one or more DHs. Examples of DPs include devices (e.g. a thermometer), applications (e.g. video streamer), Data Collectors (DCs), while DHs could be a cloud server, an edge server, and even a device with adequate storage such as a vehicle. Data storage applications exist in both DPs and DHs for jointly performing distributed data storage.

Figure 5.4-1 illustrates distributed data storage, where data from DPs is stored on a distributed data storage system consisting of multiple distributed DHs:

- Scenario 1: A DP submits its data to a single DH at a given time. For example, DP-A submits its data to DH-1. All the data from DP-A can be stored on DH-1. Alternatively, DH-1 can split the data into multiple parts, store some parts locally and transmit some parts to other DHs (e.g. DH-2). DP-A may submit its data to different DHs at different times but only one DH at any given time.

- Scenario 2: DP-B submits its original data to multiple distributed DHs. In one case, DP-B splits its data to multiple parts and submits each of those parts to a different DH. The data split may be based on volume (e.g. data blocks of same size regardless of content are submitted to different DHs sequentially) or on content (e.g. DP-B submits image files to DH-2 and text files to DH-n). In another case DP-B may submit the same data to multiple DHs in parallel for purposes such as resiliency.

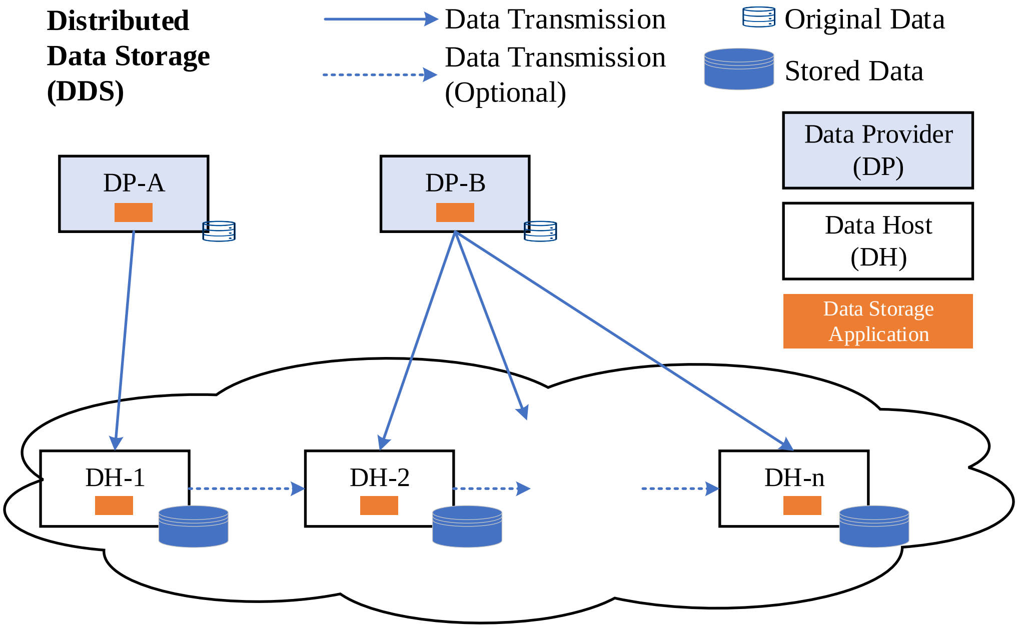

Figure 5.4-1: Distributed Data Storage

5.5 Distributed Data Sharing

Data sharing occurs between three types of data nodes: Data Providers (DPs), Data Consumers (DCSs), and Data Owners (DOs). A DP provides data, which is shared and accessed by one or more DCSs. The data in a DP could originate from one or more Data Owners (DO). A DO can share its data with DCSs through one or more DPs. Data sharing applications exist in both DPs and DCSs for jointly performing distributed data sharing.

In a distributed data sharing scenario data is provided by multiple distributed DPs. DCSs consume the data directly from DPs. Figure 5.5-1 illustrates several distributed data storage scenarios:

- Scenario 1: A DCS consumes data from different DPs at different times. For example, DCS-A consumes data from DP-1 at time t1; then, it changes to consume data from DP-2 at time t2 (e.g. a person watching a movie offered by one content provider then switching to watch another movie offered by another content provider).

- Scenario 2: A DCS simultaneously consumes data from multiple distributed DPs. For example, DCS-B consumes data from distributed DP-2 and DP-n (e.g. a financial application reading stock values from multiple stock exchanges around the globe).

- Scenario 3: A DP accesses data from other DPs. For example, DP-2 accesses data from DP-1 and DP-n. Such data is then available for DCSs to consume from DP-2 without need for such DCSs to establish consumption arrangements with DP-1/DP-n (e.g. a travel booking application that collects data from multiple airlines and allows users to book multi-leg flights operated by multiple airlines using a single booking environment).

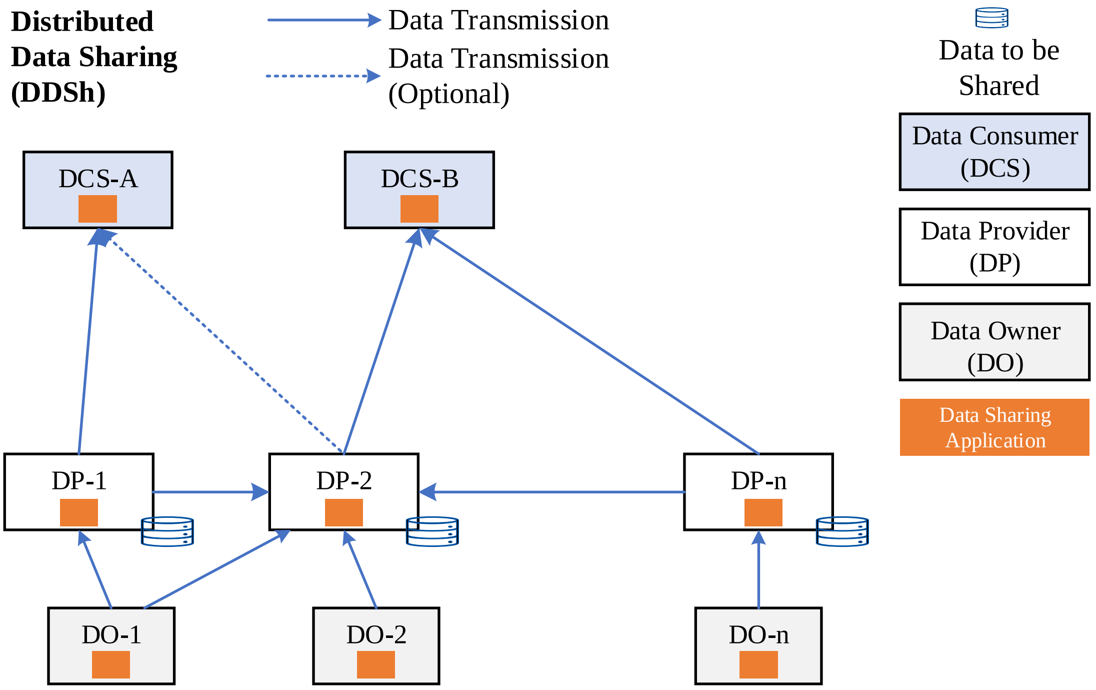

Figure 5.5-1: Distributed Data Sharing

5.6 Distributed Data Computation

Two types of data nodes are involved in distributed data computation: Data Computation Nodes (DCNs) and Data Computation Controllers (DCCs). In a distributed data computation scenario as illustrated in Figure 5.6-1, data computation is executed at DCNs, which may be coordinated by a DCC. Each DCN performs certain computation tasks over the designated data and generates computation results. DCNs may exchange computation tasks, data, and computation results with other DCNs. A DCC may assign computation tasks to DCNs and collect the computation results. Each DCN may compute local data, the received data from the DCC, and/or the data received from external data storage systems. Typical examples of distributed data computation include federated learning, federated analytics, and decentralized machine learning such as Multi-Agent Reinforcement Learning (MARL). Data computation applications exist in both DCNs and DCCs to jointly perform distributed data discovery:

- Scenario 1: A DCC and two or more DCNs collaboratively train an artificial intelligence model, referred to as federated learning. In this scenario, the DCC is the parameter server and the DCNs are the federated learning clients. The computation task at each DCN is local training, which uses local data to generate local models; in comparison, the computation task at the DCC is to aggregate local models as reported by or collected from the DCNs in order to generate the global model. The computation process at each DCN is identical but the local data is likely different.

- Scenario 2: DCNs collaboratively perform Reinforcement Learning (RL), referred to as decentralized MARL. In this scenario, there is no DCC and each DCN is a RL agent. The computation task at each DCN is RL learning process, which keeps interacting with its environment and gradually learns the optimal actions or policies. The data is the observations which the RL agent at each DCN collects from the environment. The computation result is the optimal actions or policies that the RL agent learns. Those RL agents at DCNs may exchange some information such as the observations, the learned actions, policies, etc.

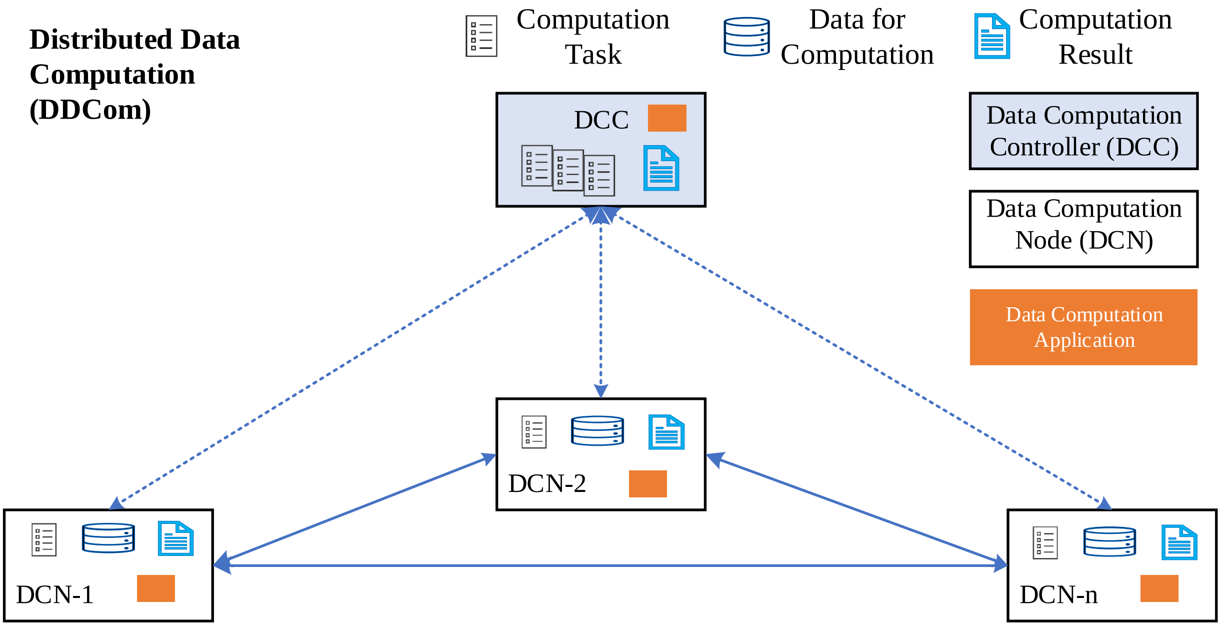
Distributed Data Computation (DDCom)
Computation Task
Data for Computation
Computation Result
DCC
Data Computation Controller (DCC)
Data Computation Node (DCN)
Data Computation Application
DCN-1
DCN-2
DCN-n

Figure 5.6-1: Distributed Data Computation

5.7 DDM Requirements

5.7.1 Introduction

Distributed data management needs the following requirements in decentralization, trust, incentivization, data provenance, data privacy, data integrity, data control and sovereignty, data management automation, etc.

5.7.2 Decentralization

In distributed data management (such as data discovery/collection, data storge/sharing, etc.), many operations are conducted in a fully distributed/parallel way and decisions are also collectively decided by multiple data nodes in the system.

[RD 1] In distributed data discovery, multiple Data Discoverers (DDs) shall collaboratively conduct data discovery in parallel at distributed Data Hosts (DHs), for serving a given data discovery task.

[RD 2] In distributed data collection, multiple Data Collectors (DCs) shall collect data in parallel and collaboratively from distributed Data Sources (DSs), for serving a given data collection task.

[RD 3] In distributed data storage, different Data Providers (DPs) shall leverage distributed DHs to upload and store their data in parallel.

[RD 4] In distributed data sharing, ad hoc data sharing relationship shall be established between a Data Provider (DP) and a Data Consumer (DCS).

[RD 5] In distributed data computation, distributed Data Computation Nodes (DCNs) shall work collaboratively to execute a given data computation task.

5.7.3 Trust

In distributed data management, different types of data nodes interact with each other to execute date operations, but they are not necessarily trusted parties. It is critical to establish trust among those data nodes in distributed data management.

[RT 1] In distributed data discovery, trust shall be established between distributed Data Hosts (DHs) and Data Discoverers (DDs) before DDs conduct discovery operations on DHs.

[RT 2] In distributed data collection, trust shall be established between distributed Data Collectors (DCs) and Data Sources (DSs) before DCs collect the desired data from DSs.

[RT 3] In distributed data storage, Data Providers (DPs) shall first establish trust relationship with the Data Hosts (DHs) before uploading and storing their data at the DHs.

[RT 4] In distributed data sharing, trust shall be first established between Data Owners (DOs) and Data Providers (DPs), between DPs and Data Consumers (DCSs), before starting any data sharing operation.

[RT 5] In distributed data computing, trustworthy interactions among data computing nodes shall be supported when multiple untrusted data computing nodes need to cooperate for a common data computing task.

5.7.4 Incentivization

In distributed data management, incentivization is an effective mechanism to motivate data nodes to participate in data operations more actively. For example, with an appropriate incentive mechanism, more distributed Data Sources will be willing to contribute their data. PDL with smart contracts naturally provides incentivization features that can be applied in a variety of different data application scenarios.

[RI 1] In distributed data discovery, Data Discoverers (DDs) shall be well incentivized so that they can actively discover data from distributed Data Hosts (DHs).

[RI 2] In distributed data collection, Data Sources (DSs) shall be incentivized so that DSs are willing to provide its data for Data Collectors (DCs) to collect.

[RI 3] In distributed data storage, certain incentivization mechanism shall be enabled so that DHs are willing to use their storage resources for storing the data for Data Providers (DPs).

[RI 4] In distributed data sharing, incentivization shall be supported to motivate Data Owners (DOs) and Data Providers (DPs) to share data to Data Consumers (DCSs).

[RI 5] In distributed data computing, incentivization shall be supported such that data computing nodes can be motivated to participate in various data computing tasks.

[RI 6] All incentives shall be agreed and documented in advance and approved by the governance to enable future accountability.

[RI 7] Incentive mechanisms shall prevent bribery and in no way give undue favour to a certain group of data nodes.

5.7.5 Data Provenance

Data provenance is to record the data origin and its evolution information throughout its lifecycle. In particular, data provenance is not only about recording the data itself during different stages, but also needs to trace which changes have been made over the data, and who made those changes, etc. In distributed data management, many data operations are involved with data exchange among multiple data nodes and data modification by multiple parties. As a result, data ownership, data status, and data quality are changing all the time. Therefore, it is critical to support data provenance for distributed data management.

[RDPV 1] In distributed data discovery, data provenance shall be supported across distributed Data Hosts (DHs) to make sure that Data Discoverers (DDs) discover data from credible DHs.

[RDPV 2] In distributed data collection, Data Collectors (DCs) shall add data provenance-related information to the collected data, if the collected data has undergone certain pre-processing/adjustment/modification compared to the original data generated by Data Sources (DSs).

[RDPV 3] In distributed data storage, different Data Providers (DPs) shall provide data provenance information to different/distributed DHs when uploading and storing data in order to support data provenance verification in the later stage.

[RDPV 4] In distributed data sharing, data provenance shall be supported to trace data origin and data sharing process.

[RDPV 5] In distributed data computing, data provenance shall be used to support trustworthy interactions among data computing nodes to avoid potential data poisoning attacks and node attacks.

5.7.6 Data Privacy

Privacy in the cyber world usually refers to private and/or personal information contained in data. The types of private information include various application data (e.g. describing user behaviours, user preference, user activity history, etc.), user personal profile (e.g. name, age, etc.), user-related identifiers (e.g. a user's smartphone number, a user's national identity, etc.). Many countries and regions have established strict regulations for privacy protection (e.g. GDPR). Distributed data management will have to meet the following requirements related to data privacy.

[RDP 1] In distributed data discovery, if target data involves user privacy, it shall not be discoverable by unauthorized Data Discoverers (DDs).

[RDP 2] In distributed data collection, Data Collectors (DCs) shall conduct the necessary data pre-processing/adjustment/modification/redaction on the raw data generated by Data Sources (DSs) if data anonymization is required to protect data privacy.

[RDP 3] In distributed data storage, when Data Providers (DPs) store their data to different distributed Data Hosts (DHs), the DHs shall only conduct authorized data replications or data movements without disclosure of private information.

[RDP 4] In distributed data sharing, data shall only be shared with authorized and intended recipients.

[RDP 5] In distributed data computing, data exchange among data computing nodes shall be compliant with existing data privacy requirements.

[RDP 6] In distributed data management, data encryption shall be supported for protecting data privacy.

5.7.7 Data Integrity

Data integrity refers to the accuracy and reliability of data. In distributed data management, it shall be ensured that data is not tampered without an authorization. PDL with its immutability feature can provide strong support for data integrity.

[RDI 1] In distributed data discovery, when discovering data from Data Hosts (DHs), Data Discoverers (DDs) shall have the capability to only discover the desired data with guaranteed data integrity.

[RDI 2] In distributed data collection, it shall be guaranteed that the data collected from the Data Sources (DSs) is not tampered during the data collection process.

[RDI 3] In distributed data storage, it shall be guaranteed that the stored data shall not be tampered.

[RDI 4] In distributed data storage, Data Providers (DPs) shall have the capability to detect if any of its data stored on DHs gets tampered.

[RDI 5] In distributed data sharing, Data Providers (DPs) shall have an agreement with Data Consumers (DCSs) indicating that the DCSs shall not tamper the data shared by DPs.

[RDI 6] In distributed data computing, data computing results shall not be tampered when the data computing results are exchanged among different data computing nodes.

5.7.8 Data Control and Sovereignty

Data control refers to how to regulate data movement and how to support data sovereignty such as data ownership and data access rights. Data ownership indicates which data node owns data and is the data owner, while data access rights indicate the privilege that allows one or multiple data nodes (not the data owners) to access the data. In distributed data management, data movement involves many data nodes and becomes more complicated. For example, a data owner may not host data itself; instead, the data owner can store the data at a data provider, from which data consumers can access the data.

[RDCS 1] In distributed data discovery, Data Hosts (DHs) shall control the discovery scope of Data Discoverers (DDs) such that if a data owner requires its data to be non-discoverable, the DDs shall not be able to discover those data.

[RDCS 2] In distributed data collection, Data Collectors (DCs) shall only be allowed to collect data from Data Sources (DSs) when DSs authorize them.

[RDCS 3] In distributed data storage, Data Hosts (DHs) shall realize necessary policies/rules in order to be compliant to the data control requirements posed by the data owner and enforce needed data access control.

[RDCS 4] In distributed data sharing, data ownership shall be guaranteed so that a Data Owner (DO) can still hold its ownership after it shares data with a Data Consumer (DCS).

[RDCS 5] In distributed data computing, the ownership of data computing results shall be managed and guaranteed.

5.7.9 Data Management Automation

Automatic operations are beneficial for data management. It not only increases the system efficiency but also reduces the burden for human intervention. Different approaches including smart contracts, policy management, and artificial intelligence can enable distributed data management automation.

[RDMA 1] In distributed data discovery, multiple Data Discoverers (DDs) shall automatically split a discovery task such that each DD is assigned to conduct discovery on one or more Data Hosts (DHs).

[RDMA 2] In distributed data collection, Data Collectors (DCs) shall automatically collect data from Data Sources (DSs) based on certain configurations such as schedules, event triggers, and/or policies.

[RDMA 3] In distributed data storage, Data Providers (DPs) and Data Hosts (DHs) shall sign a smart contract in order enforce automatic data access control.

[RDMA 4] In distributed data sharing, Data Owners (DOs) and Data Consumers (DCSs) shall sign a smart contract in order to support automatic data operations such as automatic policies enforcement and automatic payment execution.

[RDMA 5] In distributed data computing, efficient collaboration among different data computing nodes shall be supported by automatic operations such as automatic node recruitment for a common data computing task, automatic data computing task splitting and assignment, automatic reward allocation among data computing nodes, and automatic management of data computing results.

[RDMA 6] In distributed data management, smart contracts shall be supported in order to realize automatic execution of data operations (e.g. automatic payment for data sharing) among different data nodes.

6 Architectural Requirements for PDL-based DDM

6.1 Distributed Data Applications

Each distributed data management scenario has different Distributed Data Applications (DDAPPs) residing in distributed data nodes, which are involved in each distributed data management scenario. To leverage PDL for supporting distributed data management, those applications need to interact with ETSI ISG-PDL platforms, especially PDL platform services layer. DDAPP should have the following requirements:

[RDDA 1] A DDAPP shall support to discover an ETSI ISG-PDL platform including its platform service layer.

[RDDA 2] A DDAPP shall support to discover services provided by an ETSI ISG-PDL platform.

[RDDA 3] A DDAPP shall be registered to a discovered ETSI ISG-PDL platform in order to leverage services provided by the discovered ETSI ISG-PDL platform.

[RDDA 4] A DDAPP shall support to indicate other associated DDAPP to the ETSI ISG-PDL platform that it is registered to.

[RDDA 5] A DDAPP shall support to indicate its requirements and preferences on underlying DLT networks to the ETSI ISG-PDL platform that it is registered to.

[RDDA 6] A DDAPP shall support to indicate and store its application task logic to the ETSI ISG-PDL platform that it is registered to.

[RDDA 7] A DDAPP shall be assigned with a unique identifier by the ETSI ISG-PDL platform that it is registered to.

[RDDA 8] A DDAPP shall be assigned with an underlying DLT networks by the ETSI ISG-PDL platform that it is registered to.

[RDDA 9] A DDAPP shall support to store data to both external storage system and underlying DLT networks via the registered ETSI ISG-PDL platform.

[RDDA 10] A DDAPP shall support to request the registered ETSI ISG-PDL platform to create one or multiple DDM-related transactions to a designated underlying DLT networks, when the DDAPP needs to store its data or DDM-related information to DLT networks.

[RDDA 11] A DDAPP shall support to query any created DDM-related transactions from the registered ETSI ISG-PDL platform.

6.2 Platform Services Layer

To support distributed data management and corresponding distributed data applications, ETSI ISG-PDL platform services layer shall support the following requirements:

[RPSL 1] An ETSI ISG-PDL platform services layer shall be discoverable by a DDAPP.

[RPSL 2] An ETSI ISG-PDL platform services layer shall support the registration of a DDAPP.

[RPSL 3] An ETSI ISG-PDL platform services layer shall support to assign an underlying DLT network to a DDAPP that is registered to the ETSI ISG-PDL platform services layer.

[RPSL 4] An ETSI ISG-PDL platform services layer shall support to create one or multiple transactions on behalf of a DDAPP when the DDAPP requests to store its data or DDM-related information to underlying DLT networks.

[RPSL 5] An ETSI ISG-PDL platform services layer shall support the management of policies for a DDAPP for using platform services and underlying DLT networks.

[RPSL 6] An ETSI ISG-PDL platform services layer shall support the registration of an underlying DLT network.

[RPSL 7] An ETSI ISG-PDL platform services layer shall support monitoring and management of a registered underlying DLT network.

6.3 Underlying DLT Networks

To support distributed data applications for storing their data or DDM-related information to underlying DLT networks, underlying DLT networks shall support the following requirements:

[RUDLTN 1] An underlying DLT network shall be registered to an ETSI ISG-PDL platform.

[RUDLTN 2] The status of an underlying DLT network shall be monitored by ETSI ISG-PDL platforms.

[RUDLTN 3] DLT nodes of an underlying DLT network shall propagate and exchange transactions in a secure and governance-approved way.

7 PDL-based Distributed Data Management Architecture

7.1 Introduction

ETSI GS PDL 012 [1] has defined a suite of ETSI ISG-PDL platform services. This clause only focuses on services which are relevant to and/or required in order to support distributed data management.

Figure 7.1-1 illustrates an architecture for supporting PDL-based distributed data management. This architecture extends ETSI ISG-PDL reference architecture as defined in ETSI GS PDL 012 [1], but still consists of five layers:

- Application Layer: There are various distributed data applications such as DDAPP-A. Those data applications leverage services provided by PDL Platform Services Layer.
- Application Abstraction Layer: Same as in ETSI GS PDL 012 [1].
- PDL Platform Services Layer: This layer provides ETSI ISG-PDL platform services as defined in ETSI GS PDL 012 [1] and new services for supporting PDL-based DDM, referred to as DDM services. Those new DDM services can be exposed directly to DDAPPs and/or indirectly invoked by ETSI ISG-PDL platform services. For example, DDAPP-A can access those new DDM services directly. Alternatively, DDAPP-A can invoke existing ETSI ISG-PDL platform services in ETSI GS PDL 012 [1], which can call those new DDM services. Some existing ETSI ISG-PDL platform services (e.g. Application Registration) are expanded to support DDAPPs. Those expanded ETSI ISG-PDL platform services will be described in the rest of clause 7.
- DLT Abstraction Layer: Same as in ETSI GS PDL 012 [1].
- DLT Layer: This layer includes different DLT networks (e.g. DLT Network A and DLT Network B) to be leveraged by distributed data applications. A DDAPP can use one or multiple DLT networks through PDL Platform Services Layer. One DLT network can be leveraged by one or multiple distributed data applications.

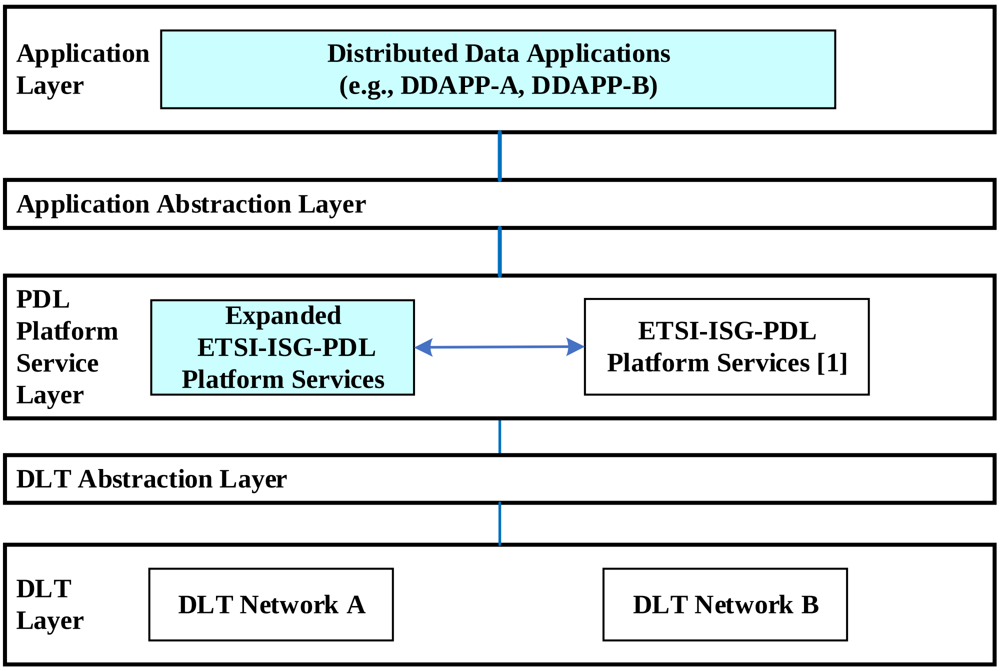
Application Layer
Distributed Data Applications (e.g., DDAPP-A, DDAPP-B)
Application Abstraction Layer
PDL Platform Service Layer
Expanded ETSI-ISG-PDL Platform Services
ETSI-ISG-PDL Platform Services [1]
DLT Abstraction Layer
DLT Layer
DLT Network A
DLT Network B

Figure 7.1-1: PDL-based Distributed Data Management Architecture

7.2 Application Registration Platform Service

ETSI GS PDL 012 [1] defines Application Registration Platform Service (ARPS). To support distributed data applications, ARPS is extended with the following new functionalities. ARPS allows a DDAPP to register itself to ETSI ISG-PDL Platform Services Layer. ARPS receives an application registration request from a DDAPP, authenticates and authorizes the application registration request, which will leverage some existing ETSI ISG-PDL platform services as defined in ETSI GS PDL 012 [1]. If the application registration request gets approved, ARPS generates a unique identifier for the DDAPP, assigns a DLT network for the DDAPP, allocates some ETSI ISG-PDL platform services (e.g. a Messaging Platform Service, a Registration Platform Service) for the DDAPP, and create a data application registration record for the DDAPP. Then, ARPS sends a response to the DDAPP indicating the unique identifier of the DDAPP and the identifier of the created data application registration record.

7.3 Registration Platform Service

Registration Platform Service (RPS) as defined in ETSI GS PDL 012 [1] is extended to support the registration of an underlying DLT network to an ETSI ISG-PDL platform. RPS is also extended to support the registration of data to an ETSI ISG-PDL platform.

An underlying DLT network provides distributed ledger capabilities and functionalities. Each DLT network may have different characteristics (e.g. ledger type, network size, adopted consensus protocol, transaction format, performance expectation, support for customizing ledger operations, etc.). RPS supports a DLT network to register itself to an ETSI ISG-PDL platform. First, the DLT network sends a network registration request to RPS indicating its characteristics. RPS authenticates and processes the network registration request. Then, RPS assigns a unique DLT ID and creates a DLT network registration record for the registered DLT network. DLT network registration records are exposed to other ETSI ISG-PDL platform services (e.g. ARPS) and/or distributed data applications. For example, Storage Platform Service (SPS) in ETSI GS PDL 012 [1] can look up DLT network registration records to find a proper DLT network, which characteristics meet data storage requirements for a distributed data application. In addition, RPS can monitor the status of DLT networks; RPS can also configure and customize some operations of DLT networks (e.g. consensus mechanisms). When a DLT Network registers to RPS, it can indicate to RPS whether it supports customizable DLT processing; as such, when there is no qualified DLT network for meeting the needs of a DDAPP (e.g. when the DDAPP is registering to ARPS), RPS can configure some operations and processing of the customizable DLT network (e.g. adopted consensus mechanism) for serving the DDAPP.

RPS also supports a Data Owner (DO) to register its data to an ETSI ISG-PDL platform with better ownership control, while allowing a Data Consumer (DC) to access the registered data. Each DO, each DC and each piece of data have their own unique identifier. RPS provides decentralized data management and access control services. A DO registers its owned data to RPS and records the data either locally, in an off-chain storage, or to an underlying DLT network. Data registration records will be maintained in underlying DLT networks. When a DC intends to access the data, the corresponding DO generates and sends an access token to the DC. The DC presents the access token to RPS. RPS authenticates the DC and verifies the access token.

7.4 Messaging Platform Service

As defined in ETSI GS PDL 012 [1], Messaging Platform Service (MPS) is expanded to support DDAPPs to use on-chain communications for exchanging their application messages through DLT networks. With MPS, exchanging application messages and recording them to distributed ledgers are conducted simultaneously. A source DDAPP sends an application message to a source MPS indicating the message shall be delivered to a destination DDAPP; the source MPS determines a destination MPS, which can reach the destination DDAPP; the source MPS sends a notification to the destination MPS indicating the incoming application message for the destination DDAPP; then, the source MPS generates a transaction containing the application message; the source MPS sends the transaction to an underlying DLT network; the transaction propagates through the underlying DLT network and is recorded in distributed ledgers; the destination MPS receives the transaction and extracts the application message from the transaction; the destination MPS delivers the application message to the destination DDAPP. For example, in distributed data collection, a data source first sends the data/message to a source MPS. Then, the source MPS discovers and determines a destination MPS, which can reach a data collector. The source MPS sends an off-chain notification to the destination MPS indicating data will be sent to the data collector via the destination MPS. After that, the source MPS transmits the data/message over an underlying DLT network to the destination MPS and the data/message is recorded in the distributed ledgers simultaneously. Finally, the destination MPS receives the data/message from the underly DLT network and forwards the data/message to the data collector.

7.5 Storage Platform Service

Storage Platform Service (SPS) defined in ETSI GS PDL 012 [1] is expanded to facilitate data nodes to store data to distributed ledgers. In the context of distributed computing (e.g. Federated Learning (FL)), FL participants as data computing nodes store FL-related data computing results (e.g. local model updates) to distributed ledgers. First, a data computing node specifies its storage requirements such as the needed data pre-processing. From the list of registered and available underlying DLT networks as maintained by RPS, SPS selects the most suitable DLT network meeting the storage requirements of the data computing node. The data computing node can also leverage RPS to interact with the underlying DLT system to request to create a customized DLT network for the data computing node. To facilitate the data computing node to submit appropriate data to SPS, SPS informs the data computing node of some instructions on the type of data to be stored to different DLT networks. Then, the data computing node submits the data to SPS according to the instructions. Finally, SPS receives the data from the data computing node, conducts data (pre)-processing if needed to reduce/avoid any duplicated data and optimize data storage, decides whether different types of data submitted by the data computing node shall be stored in the same ledger or different ledgers, and sends the processed data to the selected DLT network.

7.6 Transaction Management Platform Service

Transaction Management Platform Service (TMPS) defined in ETSI GS PDL 012 [1] is expanded to support a DDAPP to flexibly store the full or partial application data and/or the hash of the application data to distributed ledgers of an underlying DLT network in the form of transactions. First, the DDAPP sends its identifier, the application data and Transaction Creation Indications (TCIs) for handling the application data to TMPS. TCIs specify how the application data should be stored to which underlying DLT network. The DDAPP can designate an underlying DLT network or TMPS can select and determine an appropriate underlying DLT network for the DDAPP. Dependent on TCIs, the application data can be split into multiple pieces (e.g. off-chain pieces and on-chain pieces); off-chain pieces are to be stored in off-chain storage, and only their hash values and on-chain pieces will be contained in one or multiple transactions to be sent to distributed ledgers. A TCI can require TMPS to retrieve new data from an off-chain storage, combines the new data with the entire application data (or on-chain pieces) to get a combined data, creates a transaction containing the combined data, and sends the transaction to distributed ledgers. After the transaction is successful stored to distributed ledgers, it has a transaction sequence number. Finally, TMPS sends a response to the DDAPP indicating the transaction sequence number and the address for storing off-chain pieces of the application data in the off-chain storage. TMPS also supports a DDAPP to send a request to TMPS to query one or multiple specific transactions which have been created and added to distributed ledgers of an underlying DLT network; according to the request from the DDAPP, TMPS forwards the request to a distributed ledger to look up desired transactions and returns the results to the DDAPP.

7.7 Discovery Platform Service

Discovery Platform Service (DPS) defined in ETSI GS PDL 012 [1] is expanded to support a DDAPP to discover data from multiple organizations. The DDAPP issues a data discovery request to DPS for discovering desired data located in different organizations. The data discovery request contains the identifier of the DDAPP, desired data types, a list of potential organizations storing the desired data, and service fee to be deposited to a smart contract and to be paid to data discoverers from these organizations. Each organization has a data discoverer, which only has the capability for accessing or conducting discovery within its organization. To serve the data discovery request from the DDAPP, DPS uses smart contacts and PDL to solicit and coordinate multiple data discoverers from different organizations to work together for this data discovery request. To use smart contracts, DPS shall follow guidelines and specifications as defined in ETSI GS PDL 011 [2]. Each data discoverer performs a data discovery within its organization. Data discovery results from these data discoverers will be aggregated by DPS and eventually be returned back to the DDAPP by DPS. Through the data discovery request and under the control of the DDAPP, the DDAPP can request DPS to store the data discovery results in distributed ledgers, if the data discovery results are GDPR-compliant.

History

Document history		
V1.1.1	October 2022	Publication