ETSI White Paper No. #48

An Introduction of Permissioned Distributed Ledger (PDL)

1st edition – January 2022

ISBN No. 979-10-9262036-6

Authors:

Chonggang Wang (Lead), Mischa Dohler, Diego R. López, Raymond Forbes, Shahar Steiff, Tooba Faisal, Sheeba Backia Mary B., Qianren Liu, Ismael Arribas

ETSI
06921 Sophia Antipolis CEDEX, France
Tel +33 4 92 94 42 00
info@etsi.org
www.etsi.org

About the authors

Chonggang Wang, PhD

Principal Engineer, InterDigital Communications Inc., US

Chonggang Wang is a Principal Engineer with InterDigital Communications Inc. He is a Fellow of the IEEE. He is the rapporteur of the ETSI ISG PDL group report on “Federated Data Management” and group specification on distributed data management. He was the founding Editor-in-Chief (EiC) of IEEE Internet of Things Journal (2014-2016) and is currently the EiC of IEEE Networks Magazine.

Mischa Dohler, PhD

Contributed as Prof. Mischa Dohler, King’s College London; now Chief Architect at Ericsson Inc., US

Mischa Dohler is now Chief Architect at Ericsson Inc. in Silicon Valley, US. He was Professor in Wireless Communications at King’s College London, driving cross-disciplinary research and innovation in technology, sciences and arts. He is a Fellow of the IEEE, the Royal Academy of Engineering, the Royal Society of Arts (RSA), the Institution of Engineering and Technology (IET), and a Distinguished Member of Harvard Square Leaders Excellence.

Diego R. López, PhD

Telefonica S.A., Spain

Senior Technology Expert in Telefonica I+D, in charge of technology exploration and standardization activities within the GCTIO Unit. Diego is focused on applied research in network infrastructures, especially on virtualization, data-driven management, new architectures, and security. Diego chairs ETSI ISG PDL and the NOC of ETSI ISG NFV.

Raymond Forbes, PhD

Huawei Technologies Co., Ltd., UK

Pioneering blockchain introduction and deployment in the Telecommunications industry. Vice Chair of ISG PDL on distributed permissioned block-chain. Developing blockchain related ICT specifications and implementation agreements.

Shahar Steiff, MSc

AVP New Technology, PCCW Global, Hong Kong, China

Pioneering blockchain introduction and deployment in the Telecommunications industry. Developing blockchain related ICT specifications and implementation agreements. Contributor and leader in collaborative development in ETSI, CBAN, GSMA, MEF. Special interest in supply-chain management and automation.

Tooba Faisal, MSc

PhD Scholar, King’s College London (KCL), UK

Tooba Faisal is a PhD student at KCL working on Telco-blockchain. Her current research interests are designing secure smart contracts, Distributed Ledger Technology and service level agreements (SLAs). She also has a Master of Research (MRes) in Security Science from University College London, a Master of

Science (MS) in Telecommunication and Networks and a Bachelor of Science degree in Computer Engineering from Bahria University, Karachi, Pakistan. She is a KCL's delegate in the ETSI ISG PDL and rapporteur of smart contract Group Report and Specifications. She also has developed several blockchains for startups and is an IBM-trained Hyperledger Fabric developer.

Sheeba Backia Mary B., PhD

Advisory Researcher, Motorola Mobility, Deutschland

Sheeba Backia Mary B is a delegate to 3GPP, NGMN and ETSI working groups. Her key role includes developing security aspects related to communication networks, contributing to ICT standardization and specifications, leveraging blockchain security aspects and applications, etc.

Qianren Liu

China Unicom Network Communications Co., Ltd., China

Delegate and contributor to ETSI and GSMA. Promoting the application and development of blockchain technology in the telecommunications industry. Greater interest in privacy computing and data security.

Ismael Arribas

President of the Standards Commission of ALASTRIA and CEO at KUNFUD, Spain

Ismael Arribas is a Collective Entrepreneur and the CEO at KUNFUD developing independent techno-compliance and conformity assessment. He is also the Liaison officer for CEN CENELEC JTC19 and ETSI ISG PDL and Convenor at ISO TC 307 WG3 Smart Contracts and their applications.

Contents

Executive Summary

Distributed ledgers have consolidated as one of the most disruptive applications of information technology that have appeared in recent years. Their ability to store any kind of data as a consensus of replicated, shared, and synchronized digital records distributed across multiple sites, without depending on any central administrator, together with their properties regarding immutability (and therefore non-repudiation) and multi-party verifiability opens a wide range of applications, and new interaction models among those entities willing to record the transactions associated to those interactions through these ledgers. According to access permissions, there are two types of distributed ledger systems: Permissioned Distributed Ledger (PDL) and permissionless distributed ledger. In PDL, governance and corresponding access control policies are agreed by the participant community, thus better preserving privacy and allowing for a more efficient consensus protocol to achieve higher transaction speed and better energy-efficiency, three of the main concerns related to unconstrained, permissionless systems.

PDL is not a single technology but refers to a group of technologies, able to work together in different ways to support user applications. To leverage PDL technologies, a PDL reference architecture is needed, and draft ETSI GS PDL-012 currently in preparation develops a layered PDL reference architecture, based on three layers. Based on this architecture, PDL-enabled base technologies and PDL operational aspects can be considered.

Smart contracts provide mechanisms to record and execute mutual agreements among independent actors participating in the PDL. Smart contracts are executable code that live on the PDLs and inherit their properties such as transparency and immutability, being auto-executable, which means once recorded, a smart contract can perform tasks without human intervention.

PDL-based Distributed Data Management (DDM) leverages PDL to solve key issues related to trust, incentivization mechanisms, data quality control, data security and privacy in DDM scenarios. This way, DDM applications (e.g., federated learning) can be enhanced, both as a data sources or data consumers.

Offline operations deal with the aspects guaranteeing proper functionality under adverse conditions, when some PDL nodes go offline and become unreachable. These issues can be tackled and/or mitigated through various approaches such as trusted environments, offline operations, proxy mechanisms, ledger reconciliation, and PDL system monitoring and orchestration.

Inter-ledger interoperability implies the access to the records of one PDL from another PDL, and it has two modes. In unidirectional interoperability, one PDL can access data from the other PDL, what is essentially useful in data sharing applications where access is granted for specific purposes. In bidirectional interoperability, both PDLs can mutually access data and synchronize their records.

PDL can support various types of scenarios and use cases by enabling distributed trust and autonomous operations. Three especially relevant application scenarios are:

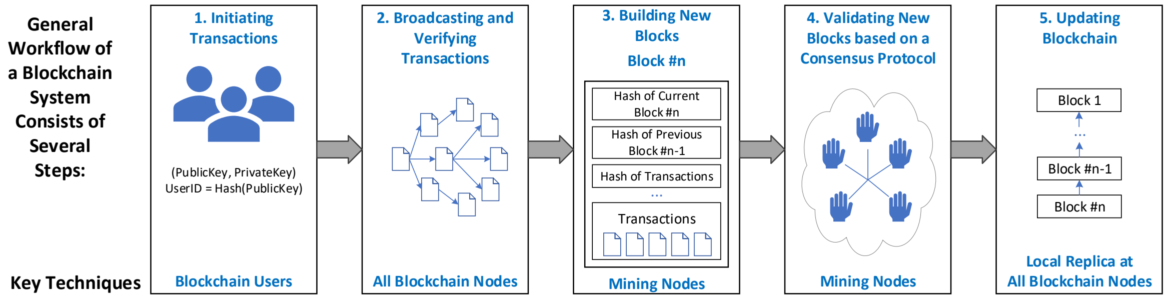
- PDL for mobile networks, enabling, for example, spectrum trading and on-demand network access. To this end, a potential new PDL Function (PDLF) may be implemented as an interworking function to interface mobile core networks and underlying PDL mechanisms.
- PDL for data sharing, ensuring the reliability of data flows results, solving data silos, supporting trusted computing environments, and applying smart contracts to maintain system security and autonomous interaction.

- PDL for AI, as a source to verify the provenance of data, models, and decisions. PDL is especially relevant for emerging distributed AI techniques, addressing several new issues such as trust, data privacy and incentivization

Finally, the field is evolving, considering new technologies to address additional application scenarios, as the case of redactable ledgers (blockmatrix), providing the integrity assurance of blockchain but allowing for controlled revision or deletion of data, or the use of payment channel networks as a solution improving throughput by applying microtransactions at a speed only impacted by the underlying network performance.

1 Introduction

As a type of Distributed Ledger Technology (DLT), Blockchain jointly leverages and builds on top of a few existing techniques such as cryptography, hashing, Merkle tree, distributed ledgers, Peer-to-Peer (P2P) networking, and consensus protocols. But it innovatively integrates them together into its workflow and enables a system which can achieve advanced features such as decentralization, immutability, transparency, and security. Figure 1 illustrates a general workflow of a blockchain system consisting of five steps: initiating transactions, broadcasting and verifying transactions, building new blocks, validating new blocks based on a consensus protocol, and updating blockchain in distributed ledgers. A transaction could contain a real cryptocurrency transaction, a digital message to be exchanged between participants (e.g., decentralized applications), a piece of data, and/or a smart contract that contains a digital contract/protocol to be leveraged by and conducted among two or more participants.


General Workflow of a Blockchain System Consists of Several Steps:
1. Initiating Transactions
(PublicKey, PrivateKey)
UserID = Hash(PublicKey)
Blockchain Users
2. Broadcasting and Verifying Transactions
All Blockchain Nodes
3. Building New Blocks
Block #n
Hash of Current Block #n
Hash of Previous Block #n-1
Hash of Transactions
...
Transactions
Mining Nodes
4. Validating New Blocks based on a Consensus Protocol
Mining Nodes
5. Updating Blockchain
Block 1
...
Block #n-1
Block #n
Local Replica at All Blockchain Nodes

Key Techniques in Each Step:					
Cryptography	Yes	Yes			
Hashing	Yes		Yes		
Merkle Tree			Yes		
Distributed Ledgers					Yes
P2P Networking		Yes	Yes	Yes	Yes
Consensus Protocol				Yes	

Figure 1: General Workflow of a Blockchain System

Although the transactions stored in distributed ledgers are tamper resistant, the access to distributed ledgers may need to be authorized. To this end, there are two types of blockchain or distributed ledger systems: Permissioned Distributed Ledger (PDL) and permissionless distributed ledger. In PDL, participants cannot freely join or leave the system like in permissionless systems but need to be authorized and governed by certain authorities (either private or consortium). In other words, governance and corresponding access control policies are critical for maintaining PDL systems. In addition, PDL may only make selected information visible to the public and will have less privacy concern than permissionless distributed ledger systems. PDL also can leverage more efficient consensus protocols such as proof-of-stake to achieve higher transaction speed and better energy-efficiency.

In 2018, ETSI established an Industry Specification Group (ISG) on PDL, referred to as ETSI ISG PDL. This ISG aims to analyze and provide the foundations for the operation of permissioned distributed ledgers, with the ultimate purpose of creating an open ecosystem of industrial solutions to be deployed by different sectors, fostering the application of these technologies, and therefore contributing to consolidate the trust and dependability on information technologies supported by global, open telecommunications networks.

As of the publication of this White Paper, ETSI ISG PDL has created multiple work items; as a result, several Group Reports (GRs) and Group Specifications (GSs) were published or are currently work-in-progress (see Table 1).

Table 1: Work Items being Developed by ETSI ISG PDL

Work Item Name	GRs/GSs	Status
Landscape of Standards and Technologies	ETSI GR PDL-001[1]	Published (March 2020)
Applicability and Compliance to Data Processing Requirements	ETSI GR PDL-002 [2]	Published (November 2020)
PDL Application Scenarios	ETSI GR PDL-003 [3]	Published (December 2020)
Smart Contracts System Architecture and Functional Specification	ETSI GR PDL-004 [4]	Published (February 2021)
Proof of Concepts Framework	ETSI GR PDL-005 [5]	Published (March 2020)
PDL Inter-Ledger Interoperability	ETSI GR PDL-006 [6]	Early draft
Research and Innovation Landscape	ETSI GR PDL-008 [7]	Published (September 2021)
PDL for Federated Data Management	ETSI GR PDL-009 [8]	Published (September 2021)
PDL Operations in Offline Mode	ETSI GR PDL-010 [9]	Published (August 2021)
Specification of Requirements for Smart Contracts Architecture and Security	ETSI GS PDL-011 [10]	Published (December 2021)
PDL Reference Architecture Framework	ETSI GS PDL-012 [11]	Stable in the public area as v0.0.3 ISG approved
PDL for Supporting Distributed Data Management	ETSI GS PDL-013 [12]	Early draft

This White Paper aims to: 1) present a brief introduction on blockchain and distributed technology; 2) summarize PDL work items and technologies that have been developed and are currently under development by ETSI ISG PDL; 3) describe selected use cases that PDL technologies can be applied to; and 4) discuss advanced DLT technologies with standardization potentials in future, such as redactable ledgers and payment channel networks. The rest of this White Paper is organized as follows:

- Section 2 describes the PDL reference architecture
- Section 3 elaborates PDL technologies as developed by ETSI ISG PDL
- Section 4 includes several use cases that PDL technologies are applicable to
- Section 5 discusses some advanced DLT technologies, and

- Section 6 concludes the White Paper.

2 PDL reference architecture

As briefly mentioned in Section 1, PDL is not a single technology but refers to a group of technologies, which could work together in many ways with different variants to enable novel PDL-based applications. In other words, different applications may rely on different DLT networks and different PDL technologies. Currently, most PDL applications are proprietary and use an integrated system where the application is developed by a specific developer and is based on a specific, prescribed, DLT technology. To make it more efficient for applications to leverage PDL technologies, a PDL reference architecture is needed. ETSI GS PDL-012 develops a layered PDL reference architecture, which consists of three layers as illustrated in Figure 2. 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.

- **PDL Application Layer:** Various PDL-based applications leverage PDL services as provided by the Platform Service Layer described below in order 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. Section 4 describes a few PDL-based application scenarios and use cases. 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.

- **PDL Platform Service Layer:** In order to support various types of applications using PDL technology, the PDL Platform Service Layer provides useful services for applications. As a result, an application could leverage services from the PDL 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 PDL Platform Service Layer could have Transaction Management Service to facilitate transaction creation in a manner transparent to a specific PDL type (i.e., a specific deployed DLT network); this is an example of layer abstraction as in essence, this Transaction Management Service can perform transaction transformation/adaptation between applications running on different PDL types to facilitate application operations in a complex environment.

- **DLT Layer:** This layer 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 uniform 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.

PDL Platform Service Layer hosts several types of services; for more details of each service, please refer to draft ETSI GS PDL-012 [11], where interface reference points between layers and among services/entities are also defined.

- The PDL Platform Services can be *Atomic services* or *Composite* services, which could be mandatory or optional. *Atomic services* are self-sufficient and do not rely on other Platform services for their proper operation, and *Composite services* use one or more other Platform service to operate.

- 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, modeling, distributed processing, resource management, service management, transaction management, etc.

- In addition, PDL Platform Service Layer has Application Specific Platform Services 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).

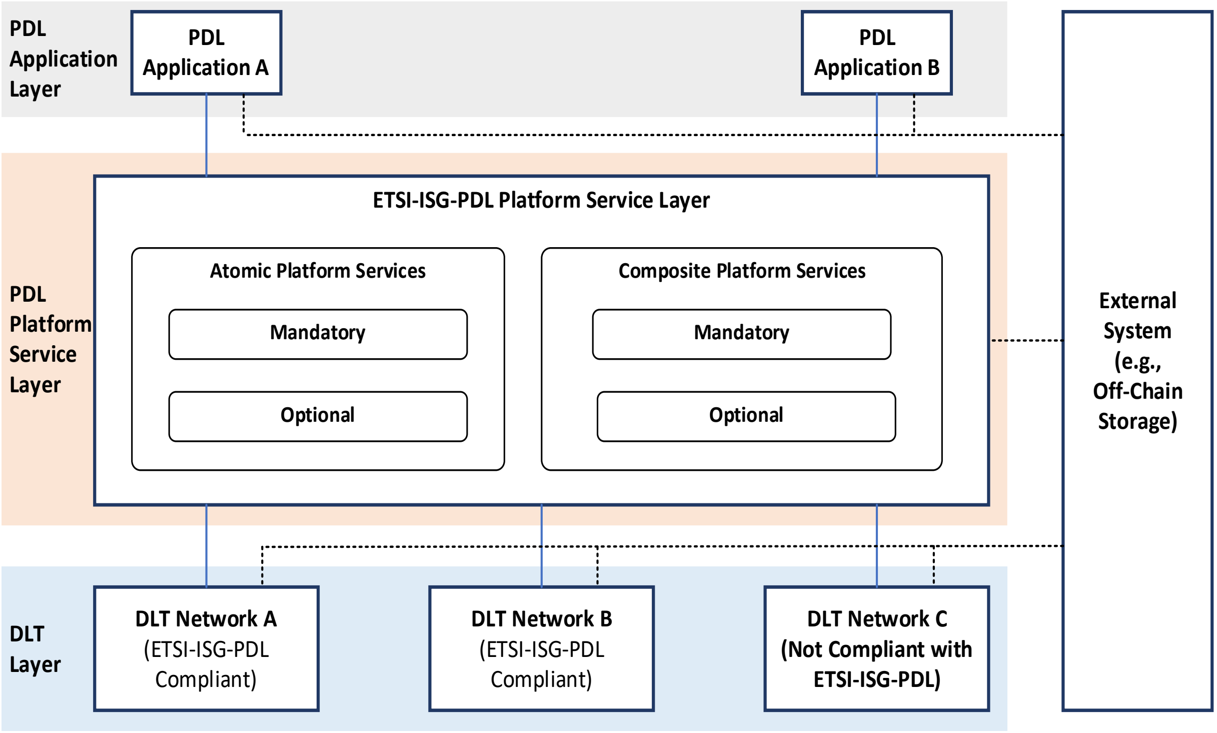

PDL Application Layer
PDL Application A
PDL Application B
PDL Platform Service Layer
ETSI-ISG-PDL Platform Service Layer
Atomic Platform Services
Mandatory
Optional
Composite Platform Services
Mandatory
Optional
External System (e.g., Off-Chain Storage)
DLT Layer
DLT Network A (ETSI-ISG-PDL Compliant)
DLT Network B (ETSI-ISG-PDL Compliant)
DLT Network C (Not Compliant with ETSI-ISG-PDL)

Figure 2: ETSI-ISG-PDL Reference Architecture (Source: Adapted from draft ETSI GS PDL-012)

3 PDL technologies

3.1 Smart contracts

PDLs on their own are static. They record the data as per the consensus of the PDL. Due to their access-control strategies and governance, PDLs are generally suitable for support business-like or enterprise applications such as auctions and supply-chain monitoring. Nevertheless, enterprises require a mechanism to record and execute their mutual agreements. Smart contracts are the solution - they are executable code that lives on the PDLs and inherits their properties such as transparency and immutability. Smart Contracts are also auto-executable, which means once recorded, a smart contract can perform tasks without human intervention. Indeed, a constructor initializes them in the first place; however, subsequent clauses are executed automatically with pre-programmed conditions. These properties require careful planning of smart contracts to avoid errors such as erroneous contracts and uncontrolled executions.

Smart contract specifications (GS-PDL 11 [10]) require the developers to adhere to the life cycle of the smart contract proposed in the Group Report ETSI GR PDL 004 [4] (See Figure 3). Following the life cycle would ensure the stepwise development of a smart contract. The stakeholders outline the requirements of a smart contract. These requirements are delivered to developers and tested at the final stage before deployment.

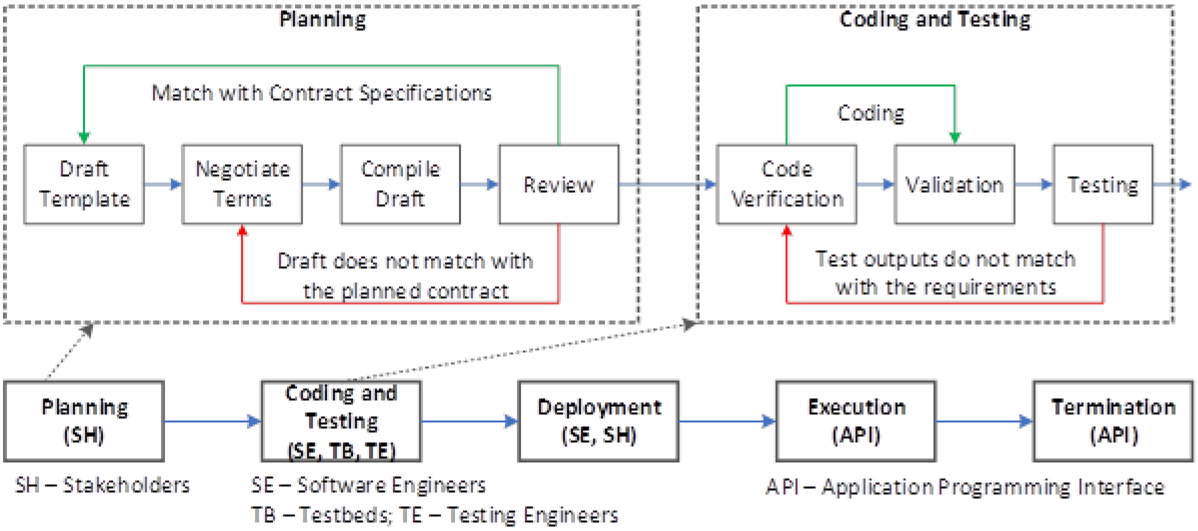

Planning
Match with Contract Specifications
Draft Template
Negotiate Terms
Compile Draft
Review
Draft does not match with the planned contract
Coding and Testing
Coding
Code Verification
Validation
Testing
Test outputs do not match with the requirements
Planning (SH)
Coding and Testing (SE, TB, TE)
Deployment (SE, SH)
Execution (API)
Termination (API)
SH – Stakeholders
SE – Software Engineers
TB – Testbeds; TE – Testing Engineers
API – Application Programming Interface

Figure 3: Smart Contract Lifecycle (Source: Adapted from ETSI GR PDL 004 [4])

A smart contract is required to give continuous monitoring after the deployment for any wrong executions. Following this stepwise life cycle would ensure security and minimize the errors in a smart contract. A smart contract is required to be tested rigorously. If a bug found during the development the smart contract is moved to a development phase again; Figure 3. Two main types of testing are specified in GS PDL 011 [10]: Pre-installation Testing and online monitoring.

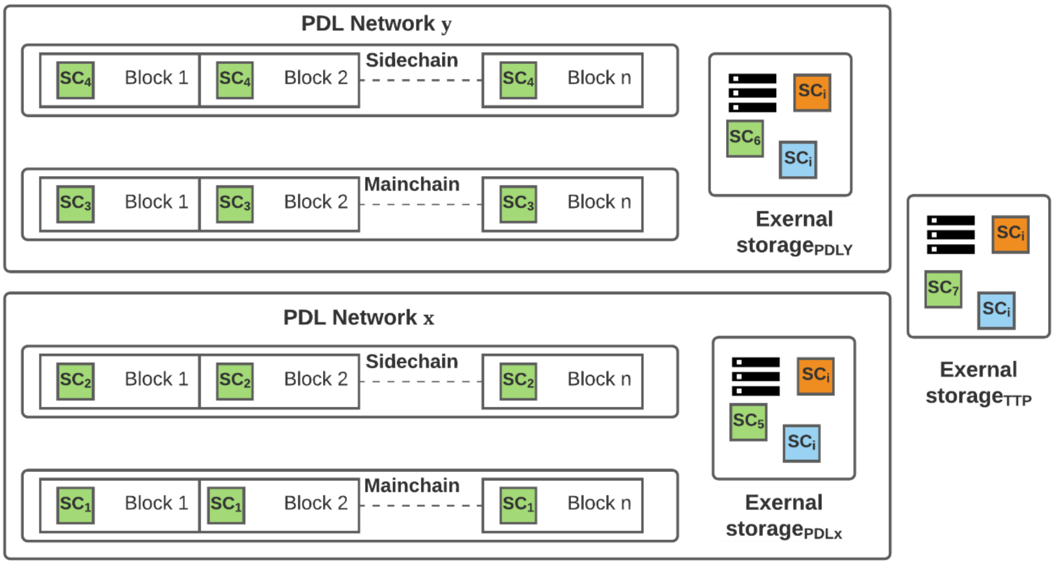


Figure 4: Modularized Smart Contract (Source: ETSI GS PDL-011 [10])

In future we envision PDLs to be in every walk of life. Future generation applications such as Machine learning (ML) and Artificial Intelligence requires extensive computations and large memory space. This is not always feasible in a distributed ledger environment because they are fast, and every node might not have this much computation power. A smart contract can be modularized and distributed across several PDLs, and external storages managed by local and external PDLs (see Figure 4 above). A smart contract module can also be offloaded to a Trusted Third Party (TTP)-managed storage. The advantages from modularized smart contracts are two-fold, 1) computation offloading, and 2) inter-operability – it enhances the sharing of contracts between different ledgers.

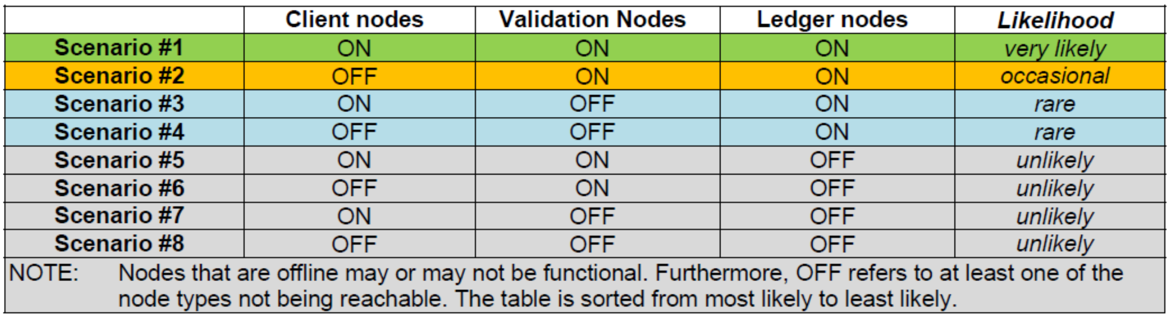
3.2 PDL offline operations

A PDL system usually consists of multiple types of PDL nodes (i.e., client nodes, validator nodes, and ledger nodes), which take different responsibilities. These nodes may be managed by one or more PDL orchestration/governance node. A client node (e.g., a device or an application) simply accesses the PDL system, for example, to submit a transaction to the PDL system. Validator nodes (e.g., an edge server) validate submitted transactions and generate blocks or other structures of a set of transactions according to certain consensus protocol. Lastly ledger nodes (e.g., a cloud server) just store validated transactions.

These PDL nodes could go offline sometime and become unreachable, referred to as PDL offline mode. PDL offline mode could be caused by original design considerations for energy conservation using duty-cycles, but also due to changing communication and networking circumstances, or incidents due to attacks or disasters. For example, a moving car as a client node may lose its connectivity to other PDL nodes due to degraded wireless channel quality.

PDL offline mode leads to various scenarios as listed in Table 2. Many technical issues may be caused such as the unavailability and insecurity of client data when a client node goes offline, the inaccessibility of ledger data when one or more ledger nodes becomes unreachable, smart contract operations, chain reconciliation, etc. These issues can be tackled and/or mitigated through various approaches such as trusted environments, offline operations, proxy mechanisms, ledger reconciliation, and PDL system monitoring and orchestration. For example, proxy nodes could be pre-selected for a validator node; when the validator node becomes offline, its proxy nodes can take over PDL operations on its behalf.

Table 2: Possible PDL Scenarios due to PDL offline Mode (Source: ETSI GR PDL-010 [9])



	Client nodes	Validation Nodes	Ledger nodes	*Likelihood*
Scenario #1	ON	ON	ON	*very likely*
Scenario #2	OFF	ON	ON	*occasional*
Scenario #3	ON	OFF	ON	*rare*
Scenario #4	OFF	OFF	ON	*rare*
Scenario #5	ON	ON	OFF	*unlikely*
Scenario #6	OFF	ON	OFF	*unlikely*
Scenario #7	ON	OFF	OFF	*unlikely*
Scenario #8	OFF	OFF	OFF	*unlikely*
NOTE: Nodes that are offline may or may not be functional. Furthermore, OFF refers to at least one of the node types not being reachable. The table is sorted from most likely to least likely.				

3.3 PDL for distributed data management

In future ICT systems, more data will be generated and stored in distributed places for example to protect data privacy and reduce data transmission overhead that could be very high in current cloud-based centralized data management. In such distributed data management scenarios, there are many participants from different organizations, which necessarily trust each other or may not be always willing to contribute their data management capabilities. This leads to several key technical issues in distributed data management: trust among participants, incentivization mechanism for participants, data quality control, data security and privacy.

PDL technologies help to solve and/or mitigate the above issues. For example:

- PDL can be leveraged to build trust relationships among untrusted participants/parties/organizations involved in federated data management.
- Smart contracts can be leveraged as an effective mechanism to incentivize participants/parties/organizations to participate in federated data management and to enable autonomous interactions among them.
- PDL can be leveraged not only for recording data, but also a mechanism to propagate/transmit data among participants/parties/organizations involved in federated data management.

In the context of PDL-based Distributed Data Management (DDM), there are two separate systems, namely PDL system and DDM applications. To leverage PDL to solve key issues as described in previous clause and eventually enable PDL-based DDM, these two systems need to interact and interwork with each other. Figure 5 illustrates a general proxy-based solution to interwork DDM system and PDL system, where DDM-PDL Proxy (DPP) is included as a logical entity to connect both systems. Via DPP, DDM applications (e.g., federated learning), which could be a data source or a data consumer, can access PDL systems, for instance, to store DDM-related data (e.g., operation records) to a PDL chain. DPP can provide the following functions:

- find appropriate PDL chains from PDL system for a DDM application based on its requirements
- interact with PDL system on behalf of a DDM application
- buffer and send requests (e.g., to create a transaction) from a DDM application to PDL system
- buffer and forward notifications and/or responses from PDL system to a DDM application and
- knows how to interact to DDM applications and how to talk to PDL systems (e.g., ledgers).

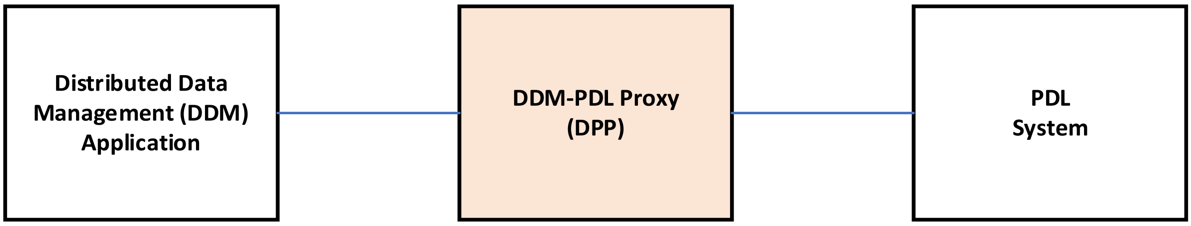


Figure 5: PDL for Distributed Data Management (Source: Adapted from ETSI GR PDL 009 [8])

3.4 Inter-ledger interoperability

Inter-ledger Interoperability is the means to access the records of one PDL from another PDL. Interoperability is helpful in several ways, such as solving the scalability problems of PDLs and the future adoption of PDL, where there will be several PDLs running by organizations around the world. However, this comes with several challenges in terms of, for example, security and ledger compatibility. Based on harmonized definitions for interoperability in ISO/IEC 17788:2014 [19], inter-ledger interoperability for exchanging information between different PDLs can be conducted in two dimensions:

- **Unidirectional Interoperability:** In unidirectional interoperability, one PDL can access the record from the other PDL but otherwise is not possible. This is essentially useful in data sharing applications where data access is given to other organizations only for specific purposes, most of the time read-only and for a limited time. For instance, in the case of medical records, the doctors would require access to people's health records but are not allowed to synchronize their ledgers with the record. The advantage of using PDLs instead of traditional databases is that they eliminate the risk of single-point-of-failure.

- **Bidirectional Interoperability:** In bidirectional interoperability, both the PDLs can access each other's data and synchronize their ledgers with the records. For example, if two hospital trusts are running their ledgers and patients move from one trust to a different trust, medical records between two PDLs will require to be synchronized. Because Bidirectional Interoperability is a two-way process, both ledgers update their nodes with the incoming data from other PDL; there are several considerations, such as compatibility between different ledgers in terms of latency and technology. The currently available technology is still in its infancy and requires improvements to enable successful bidirectional interoperability.

4 Scenarios and use cases

PDL can support various types of scenarios and use cases, for example, to enable distributed trust and autonomous operations based on smart contracts. ETSI GS PDL-005 [5] specifies the process for developing Proof-of-Concept (PoC) projects to demonstrate PDL use cases. Several completed and ongoing PoC projects include (please refer to https://pdlwiki.etsi.org for more details):

- Intelligent Decentralized Internet Infrastructure (DII) - Completed;
- Secure Marketplace for Access to Ubiquitous Goods (SMAUG) - Completed; and
- Timeless in Metaverse Environment based on Edge networks (TIME) – Ongoing

4.1 PDL for mobile networks

PDL can be leveraged to benefit various mobile networks and vertical services, for example, to enable spectrum trading and on-demand network access. To this end, a potential new PDL Function (PDLF) may be implemented as an interworking function to interface mobile core networks and underlying PDL networks (see Figure 6). PDLF provides some PDL services as defined in draft ETSI GS PDL-012 [11]. PDLF being an interworking function can facilitate the interactions between PDL participants (e.g., applications at end-devices and mobile network control plane functions) and various PDL networks. For instance, PDLF can offer the blockchain service enabler functionalities to enable decentralized identification and authentication of the User Equipment (UE) requesting service or network access by verifying the identity and corresponding service specific verifiable credentials stored on a ledger.

As an example, PDL can largely benefit the identification and authentication service to enable a more privacy-protected digital identification and authentication (i.e., using a digital identifier that can be self-sovereign and user-controlled) for seamless on-demand network access and service provision.

- Currently, the devices which do not have network access credentials and that needs initial network access for onboarding are configured with default credentials by the device manufacturer and those credentials are stored in a default credential server. The network operator having a service level agreement with the device manufacturer can allow the device to onboard the network based on the authentication by the default credential server. On a successful onboarding, the device can be provisioned with the actual network subscription credentials to allow the device to connect with the network using the actual network subscription information.
- Similarly, most of the evolving on-demand network access use cases requires dynamic user/device network access credentials configuration at the UE side, where we cannot expect all devices to come with a static default configuration. In such scenarios, the principle of electronic Identification, Authentication and Trust Services (eIDAS) framework can be used by the user and service provider to allow the user to create digital identifier and corresponding service specific access credentials in real-time to enable onboarding to the network. The digital identifier can be a Decentralized Identifier (DID) as defined by W3C "Decentralized Identifiers (DIDs) v1.0, Core architecture, data model, and representations" [13] and they are unique identifiers created by an entity in a self - sovereign way, which can be stored along with network/service access credentials in a PDL to enable network access related to the operator service or 3rd party service respectively. The information flow of the verifiable claim(s) generation and use "eIDAS Supported Self-Sovereign Identity" [14] is depicted in Figure 7. eIDAS solutions can be leveraged to increase the security of digital business services and processes with effective customer identification (see EIDAS solutions overview [15]). European Blockchain Services Infrastructure (EBSI) with its generic design supports multiple storage strategies (i.e., for DID document that specifies proof purposes, verification methods and service end points) such as on Full-On-Chain or Hybrid On-chain/Off-chain to accommodate different use case legal requirements.
- A few example scenarios that require on-demand network access can include localized network service at large sports event or cultural fest, network operator providing Operator/3rd party service over a hosting network for which the user has no subscription credentials, a user landing at an airport for business visit requires a short-term network subscription with a new network, etc.

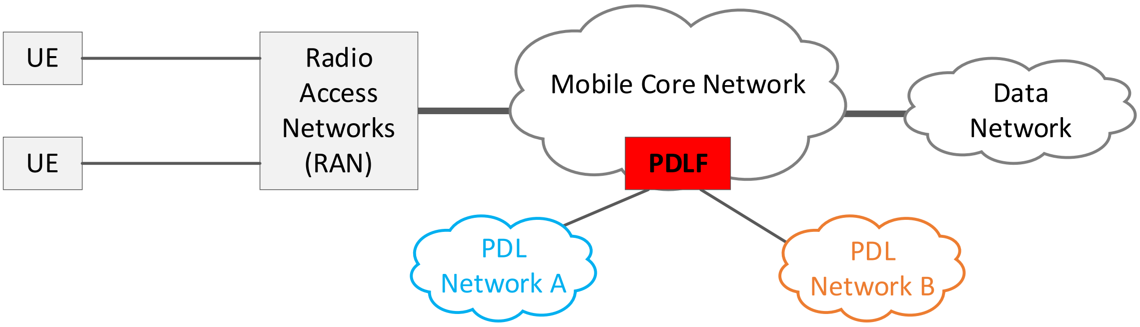


Figure 6: PDL Function in Mobile Core Networks (Source: Adapted from ETSI GR PDL 010 [9])

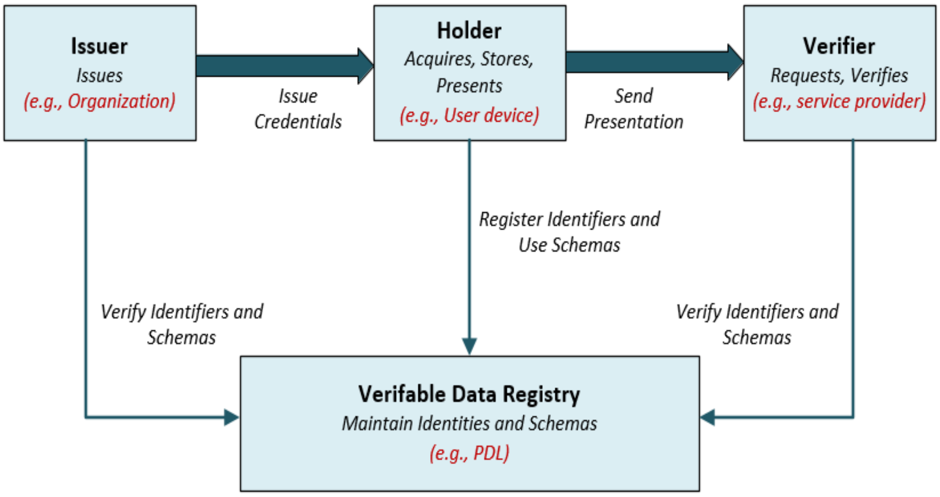


Figure 7: Information Flow of the Verifiable Claim(s) Generation and Use (Source: eIDAS [15] with PDL examples)

4.2 PDL for data sharing

PDL can solve the problem that private data cannot be shared for training. For example, different institutions need to gather all data together for training. As the carrier for storing data results, PDL ensures the reliability of training results, solves data silos to a great extent, realizes centralized training in trusted computing environment, and uses smart contracts to maintain system security and autonomous interaction.

Figure 8 illustrates a trusted computing and data sharing use case based on PDL. There are four parties in this use case: Data Demander (DD), Data Provider (DP), Data Trainer (DT) and Data Warehouse (DW).

- Data Demander: the purchaser of the service and results that are provided by the data warehouse who processes the raw data using corresponding algorithm.
- Data Provider: the owner of the data resources or capabilities. Data Provider are responsible for the availability, correctness, and accuracy of the data. Moreover, data owners should perform request authorization and data assignment. Data provider could have several sub roles, e.g., system maintainer, person liable for the data, etc.
- Data Trainer: the direct purchaser of the data resources or capabilities in the data warehouse and should be the provider, and moreover could be the developer or compiler of an algorithm.
- Data Warehouse: the platform where data transactions and operations are performed. Data warehouses also need to provide necessary techniques such as Trusted Execution Environment (TEE), registration, dispatch, coordination, maintenance, and other necessary services. Data Warehouse could have several sub roles, for example: system developer, system maintainer, system administrator, user administrator, etc.
- Blockchain Platform: the platform provides necessary DLT functions and services.

Figure 8 also illustrates the following example steps:

1. DD, DP, DT, and DW should register a workspace on the blockchain platform. During registration, identity information will be clarified, and to confirm compliance with smart contracts between the four entities. While the registration is done, a workspace will be made available to all the specified users. Then, DP should upload the encrypted raw data sets to the DW. DW should calculate and generate a corresponding data summary file and register the file to blockchain through Distributed Data Management of PDL (DDM-PDL).

2. DD publicizes a certain demand to the data trainer. The request should give out the estimate of data needed, result expected, algorithm chosen, etc. DT should analyze the exact demand of data and algorithm resources according to the request. The exact demand order should in detail include the requirement and framework of algorithm, required data type and data size of this order, training method the of the algorithm, necessary authorization information, Information of DP and so on. The exact demand order should be sent to the DP whilst the DP have the raw data in accordance with DT's exact demand order.

3. DP reviews the received exact demand order, checking the availability and compliance of both DT's request and DP's resources. Once the order is authorized, DP upload corresponding encrypted raw data set into TEE according to the provisions of smart contract. In the meantime, DT check the logic, grammar and compliance of the algorithm, and upload checked algorithm into TEE.

4. Once uploading is done, TEE, an appurtenant hardware service provided by DW, decrypts and processes DP's raw data sets using DT's algorithm and returns a final training result to DW.

5. DW summarizes the training result, collects and generates a summary file, and uploads the file to the blockchain platform as the certification of the result. Once the blockchain platform finishes file registration, DW returns the result to DT.

6. DT check the facticity of the results according to the summary file on blockchain. Once the consistency is authorized, DT requires DW to discard the algorithm in this order and provides the result to DD. Relying on smart contracts, DD pays service fees, and DT and DP get corresponding remuneration. The order is complete. DD obtains required result from DP. PDL can guarantee data security and process automation on the premise of data sharing.

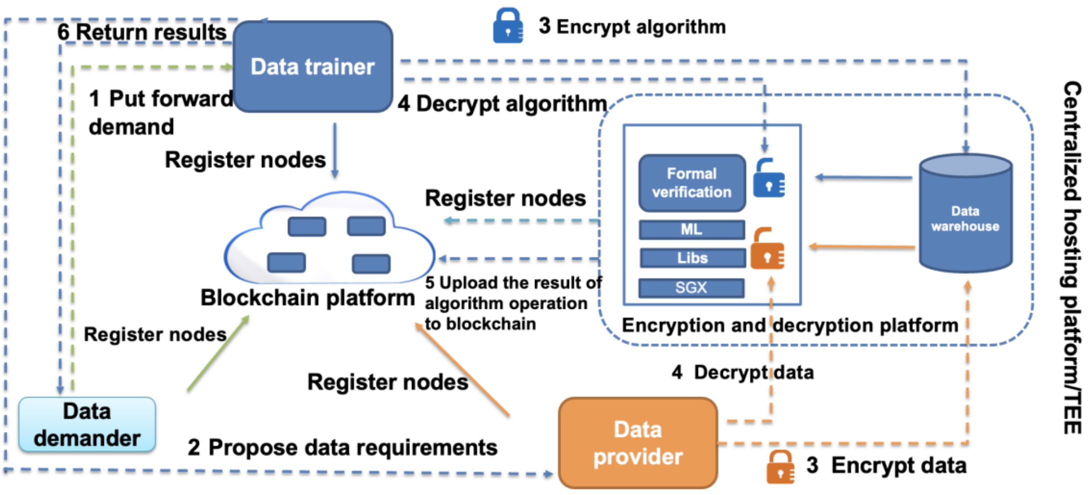


Figure 8: Trusted Computing and Data Sharing Scheme based on Blockchain

4.3 PDL for Artificial Intelligence

PDL can be leveraged for both traditional centralized Artificial Intelligence (AI) and emerging distributed AI (e.g., multi-agent reinforcement learning, distributed machine learning, Federated Learning (FL)).

For centralized AI, the mapping between training data and the generated AI model can be recorded in distributed ledgers to provide immutable training history. Furthermore, ledgers can also be leveraged to maintain temporary training results from each training epoch and in turn facilitate AI traceability and explainability. After an AI model is generated, smart contracts can be leveraged to enable AI marketplace (e.g., AI model sharing).

Several new issues such as trust, data privacy and incentivization emerge in distributed AI. As a distributed AI technology, federated learning was to implement a distributed AI model training process by multiple FL participants while still ensuring data privacy, security and legal compliance. Figure 9 illustrates a general use case of FL-enabled smart city and smart transportation.

- In smart city applications, many cameras will be deployed on streets and generate continuous data or data streams. These urban camera data can be used to train an AI model for urban environmental monitoring and predicting. However, uploading all camera data to cloud could be cumbersome or unrealistic. Accordingly, FL is a more feasible and efficient method.
- Similarly, in smart transportation applications, there will be a large number of vehicles driving on the road, and each vehicle will generate massive real-time driving data. These data can be trained to generate many AI models (e.g., to predict which road sections or during which time periods vehicles are most likely to have poor driving behavior/performance). However, these data are not only large in quantity, but also contain personal privacy information; as a result, it is unwise or inefficient to upload these data to a cloud for centralized processing/training as in traditional AI. FL can be applied in this use case such that a global ML model can be jointly trained by vehicles without uploading driving data from vehicles to cloud.

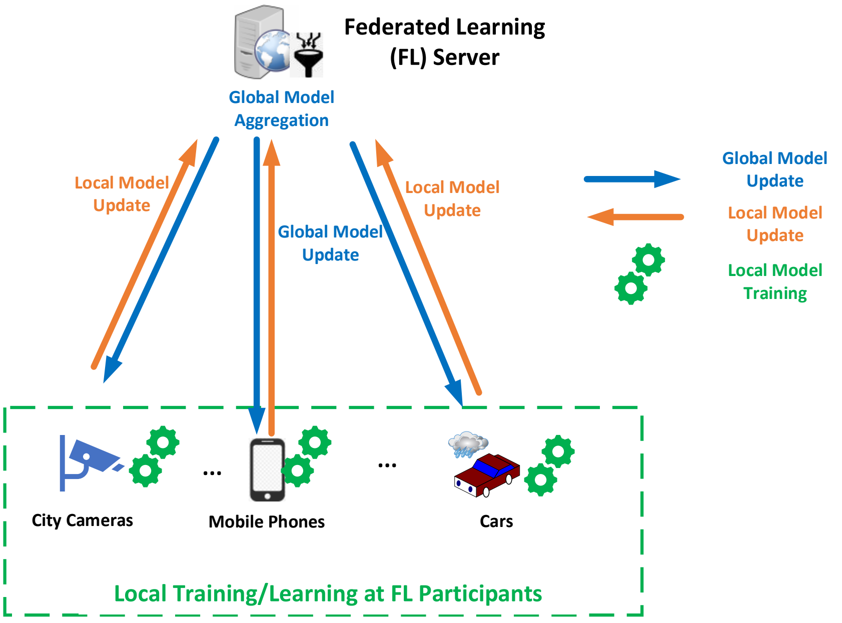


Figure 9: Federated Learning in Smart City and Smart Transportation (Source: ETSI GR PDL-009 [8])

However, FL inherits some issues such as trust, incentive mechanisms, and data quality, which are all related to and/or caused by distributed FL participants. For example, many FL participants are not from the same organization and do not trust each other, which makes effective collaboration and coordination between them difficult especially in a fully distributed scenario. In addition, an FL participant could have useless and even malicious local data, which cannot help training a good local model. PDL technologies help to solve and/or mitigate the above issues. For example, local models can be stored in the ledger for future traceability and explanation purposes, using PDL services as defined as a part of PDL Platform Service Layer in draft ETSI GS PDL-012 [11]. Also, smart contracts can be leveraged to encourage FL participants to actively cooperate and contribute their local data and learning capabilities; for example, a smart contract be created for a FL task, based on which, an FL participant can collect rewards automatically after it makes contributions to FL training process.

5 Advanced DLT technologies

5.1 Redactable ledgers

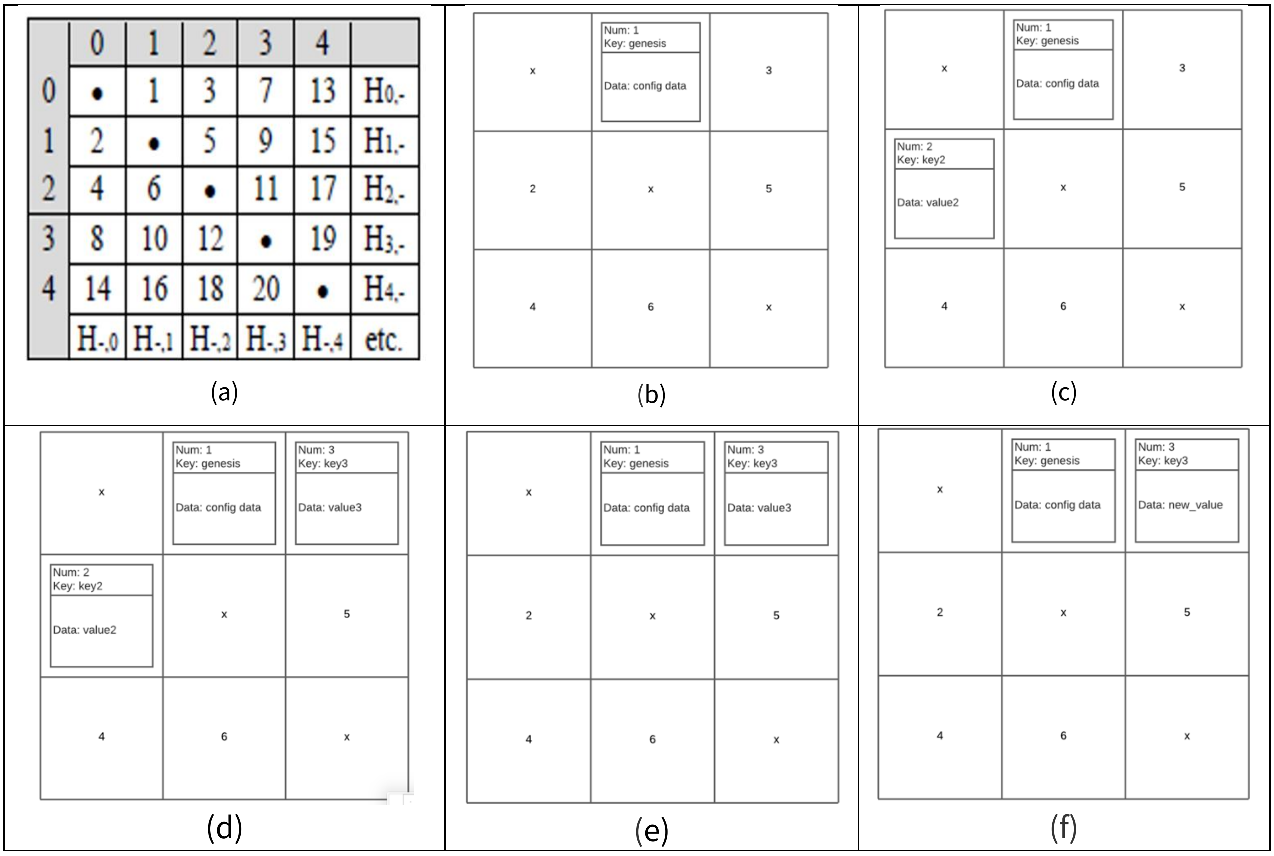
The integrity protection provided by blockchains was designed to solve the problem of double spending in cryptocurrencies. But this integrity protection is an attractive property for many applications beyond digital currency, and system designers have sought to employ blockchains in many different domains, such as logistics and e-commerce. Blockchains provide a strong mechanism to ensure that data blocks have not been altered, by using a chain of hashed data values, so that it is impossible to delete or change a block without disrupting all other blocks. As such, a blockchain is considered *immutable*, as it is not possible to change any bit without requiring the entire chain to be rebuilt, which is generally infeasible for a large blockchain.

The added trust of distributed ledgers is a valuable feature, providing greatly simplified auditability and verification of actions among multiple parties in applications such as supply chain and others. However, this immutability property also makes conventional blockchains hard to use in many distributed system applications. Many privacy requirements, such as those in the EU General Data Privacy Regulation (GDPR), allow users to have private data deleted at their request. The immutability property may make a blockchain solution impractical when privacy rules are required. In other words, redactable ledgers are useful for some applications. Proposals for redactable DLT include a blockchain based on chameleon hash as described in “Redactable Blockchain” [16] and data block matrix (or blockmatrix) from “A Data Structure for Integrity Protection with Erasure Capability” NIST whitepaper [17].

Blockmatrix is a new form of DLT, which provides the integrity assurance of blockchain but allows for controlled revision or deletion of data. This is an essential property for using DLT in applications that must support privacy requirements for deletion of private data at a user's request. Blockmatrix therefore *extends the range of application for blockchain solutions* by solving the conflict between privacy regulations and blockchain and allowing block edits or deletions for privacy requirements or for exception management.

To make the data blockmatrix solution usable and practical in real-world applications, an implementation in Hyperledger Fabric is being developed. Users will be able to configure individual Fabric channels to use either a blockmatrix or a blockchain. Once a channel is configured to use a blockmatrix, the integration is designed to be transparent to the Fabric user. This design focuses on adapting the block storage mechanisms to be compatible with the functionality of a blockmatrix. For more information, see: https://csrc.nist.gov/Projects/enhanced-distributed-ledger-technology

Figure 10 below shows (a) the basic blockmatrix structure. Blocks are added in the order indicated, such that the number of blocks above and below the diagonal is balanced. Hashes for each row and each column are shown in the last row and column, so that every block is protected by two hash values. Frames (b) through (f) show processing in Hyperledger Fabric, progressing through the following sequence in Hyperledger: (b) Initial block; (c) AddBlock(2, “key2”, “value2”); (d) AddBlock(3, “key3”, “value3”); (e) DeleteBlock(2); (f) UpdateBlock(“key3”, “new_value”).



	0	1	2	3	4	
0	•	1	3	7	13	$H_{0,-}$
1	2	•	5	9	15	$H_{1,-}$
2	4	6	•	11	17	$H_{2,-}$
3	8	10	12	•	19	$H_{3,-}$
4	14	16	18	20	•	$H_{4,-}$
	$H_{-,0}$	$H_{-,1}$	$H_{-,2}$	$H_{-,3}$	$H_{-,4}$	etc.

Figure 10: Blockmatrix Structure and Hyperledger Fabric Implementation

5.2 Payment channel networks

Payment Channel Network is a solution for improving blockchain system throughput as described in “High Throughput Cryptocurrency Routing in Payment Channel Networks” [18]. As shown in Figure 11, a Blockchain Node (BCN) can issue off-chain micropayments to other BCNs directly or use another BCN as a proxy. Such off-chain micropayments do not need to be validated and there is no consensus process; as a result, BCNs can exchange micropayments at a high speed that is only impacted by physical networking speed. Only when two BCNs need to execute and confirm their micropayments, they start to generate one on-chain regular transaction, which can contain multiple unconfirmed micropayments. In fact, the payment channel network is an overlay network representing micropayment relationships among BCNs. Although the native transaction speed of the blockchain network has not been changed, BCNs and users can issue as many micropayments via payment channel network as they want. As a result, the blockchain system throughout and scalability are improved. Effective interactions between payment channel network and blockchain network are needed. In addition, routing micropayment transactions within payment channel network is a critical issue.

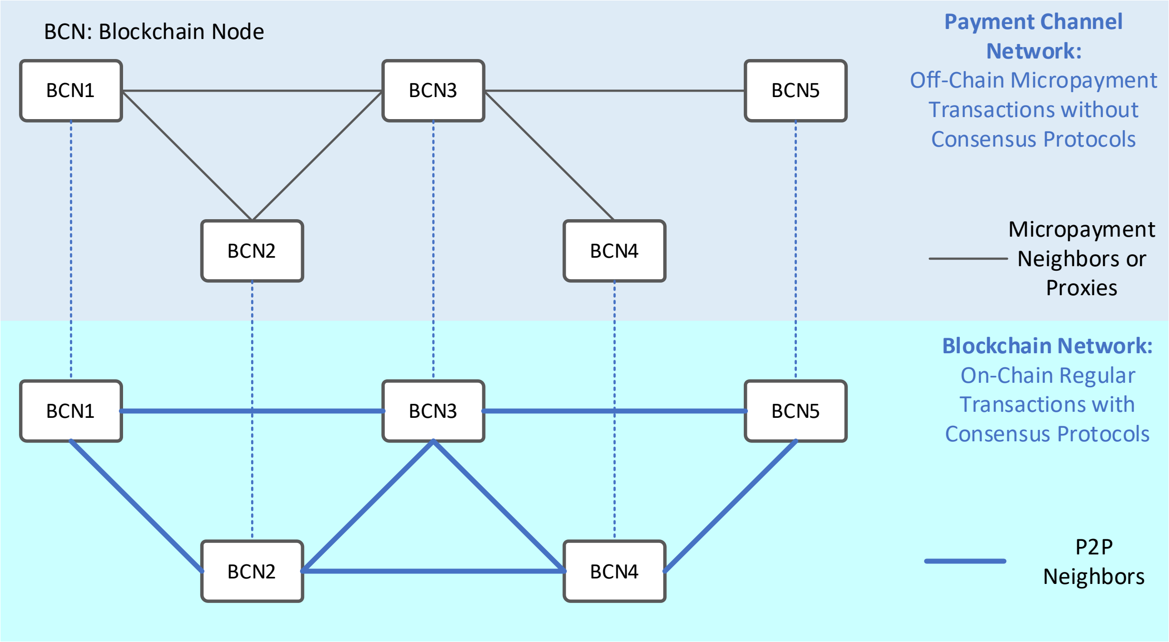

BCN: Blockchain Node
BCN1
BCN3
BCN5
BCN2
BCN4
Payment Channel Network:
Off-Chain Micropayment Transactions without Consensus Protocols
Micropayment Neighbors or Proxies
Blockchain Network:
On-Chain Regular Transactions with Consensus Protocols
BCN1
BCN3
BCN5
BCN2
BCN4
P2P Neighbors

Figure 12: Payment Channel Networks

6 Conclusions

PDL provides a permissioned distributed ledger technology, where the access to ledgers needs to be authenticated, authorized and governed, for instance, by some access control policies and rules. Different PDL-based decentralized applications may leverage different types of PDL networks. ETSI ISG PDL defines a common PDL reference architecture, where PDL Platform Service Layer is proposed to coordinate the interactions between applications and underlying PDL networks. For example, PDL Platform Service Layer can monitor offline PDL nodes and orchestrate appropriate offline operations to mitigate and resolve potential impacts from any PDL nodes becoming offline. Smart contracts will play a critical role in PDL, but the security through the entire lifecycle of a smart contract needs to be guaranteed. PDL Platform Service Layer can also facilitate the management of smart contracts from security and other aspects. PDL Platform Service Layer also manages the inter-ledger interoperability so that messages can be exchanged efficiently between PDL networks. PDL can be leveraged to support various scenarios (e.g., mobile networks, distributed data sharing and management, and artificial intelligence and machine learning) by providing immutable ledgers, distributed trust, and incentivization and autonomous operations based on smart contracts.

In addition, advanced DLT technologies such as redactable ledgers and payment channel networks are emerging to enhance the performance of PDL networks. Those advanced DLT technologies also bring some new issues for further study. For example, whether and how can PDL reference architecture including PDL Platform Service Layer effectively and efficiently leverage emerging DLT technologies to empower even more novel decentralized applications.

Acknowledgement

We acknowledge the assistance of Rick Kuhn and Joshua Roberts from the National Institute of Standards and Technology (NIST) on Section 5.1.

Annex A: List of abbreviations

AI	Artificial Intelligence
DDM	Distributed Data Management
DID	Decentralized Identifier
DLT	Distributed Ledger Technology
eIDAS	electronic Identification, Authentication and Trust Services
FL	Federated Learning
GDPR	General Data Privacy Regulation
GR	Group Report
GS	Group Specification
ISG	Industry Specification Group
ML	Machine Learning
P2P	Peer-to-Peer
PDL	Permissioned Distributed Ledger
PDLF	PDL Function
SC	Smart Contract
TEE	Trusted Execution Environment

Annex B: References

1. ETSI GR PDL 001 V1.1.1 (2020-03); Permissioned Distributed Ledger (PDL); Landscape of Standards and Technologies (http://www.etsi.org/deliver/etsi_gr/PDL/001_099/001/01.01.01_60/gr_PDL001v010101p.pdf)

2. ETSI GR PDL 002 V1.1.1 (2020-11); Permissioned Distributed Ledger (PDL); Applicability and compliance to data processing requirements (http://www.etsi.org/deliver/etsi_gr/PDL/001_099/002/01.01.01_60/gr_PDL002v010101p.pdf)

3. ETSI GR PDL 003 V1.1.1 (2020-12); Permissioned Distributed Ledger (PDL); Application Scenarios (http://www.etsi.org/deliver/etsi_gr/PDL/001_099/003/01.01.01_60/gr_PDL003v010101p.pdf)

4. ETSI GR PDL 004 V1.1.1 (2021-02); Permissioned Distributed Ledgers (PDL); Smart Contracts; System Architecture and Functional Specification (http://www.etsi.org/deliver/etsi_gr/PDL/001_099/004/01.01.01_60/gr_PDL004v010101p.pdf)

5. ETSI GS PDL 005 V1.1.1 (2020-03); Permissioned Distributed Ledger (PDL); Proof of Concepts Framework (http://www.etsi.org/deliver/etsi_gs/PDL/001_099/005/01.01.01_60/gs_PDL005v010101p.pdf)

6. Draft ETSI GS PDL 006 V0.0.9 (2021-09); PDL Inter-Ledger Interoperability - Early Draft

7. ETSI GR PDL 008 V1.1.1 (2021-09); Permissioned Distributed Ledger (PDL); Research and Innovation Landscape (http://www.etsi.org/deliver/etsi_gr/PDL/001_099/008/01.01.01_60/gr_PDL008v010101p.pdf)

8. ETSI GR PDL 009 V1.1.1 (2021-09); Permissioned Distributed Ledger (PDL); Federated Data Management (http://www.etsi.org/deliver/etsi_gr/PDL/001_099/009/01.01.01_60/gr_PDL009v010101p.pdf)

9. ETSI GR PDL 010 V1.1.1 (2021-08); PDL Operations in Offline Mode (http://www.etsi.org/deliver/etsi_gr/PDL/001_099/010/01.01.01_60/gr_PDL010v010101p.pdf)

10. ETSI GS PDL 011 V0.0.3 (2021-09); PDL: Specification of Requirements for Smart Contracts' Architecture and Security (https://www.etsi.org/deliver/etsi_gs/PDL/001_099/011/01.01.01_60/gs_PDL011v010101p.pdf)

11. Draft ETSI GS PDL 012 V0.0.3 (2021-12); Permissioned Distributed Ledger (PDL); Permissioned Distributed Ledger; Normative Reference Architecture (https://docbox.etsi.org/ISG/PDL/Open/0012_Ref_Arc_Framwk/PDL-0012_Ref_Arc_Framwkv003.docx) ISG approved

12. Draft ETSI GS PDL 013 V0.0.1 (2021-12); Permissioned Distributed Ledger (PDL); PDL for Supporting Distributed Data Management (Draft)

13. Decentralized Identifiers (DIDs) v1.0, Core architecture, data model, and representations, (https://www.w3.org/TR/did-core/).

14. eIDAS Supported Self-Sovereign Identity, 'https://ec.europa.eu/futurium/en/system/files/ged/eidas_supported_ssi_may_2019_0.pdf'.

15. eIDAS solutions overview, 'http://ec.europa.eu/newsroom/dae/document.cfm?doc_id=53854'.

16. G. Ateniese, B. Magri, D. Venturi and E. Andrade, “Redactable Blockchain – or – Rewriting History in Bitcoin and Friends,” 2017 IEEE European Symposium on Security and Privacy (EuroS&P), 2017, pp. 111-126, doi: 10.1109/EuroSP.2017.37.
17. D. R. Kuhn, “A Data Structure for Integrity Protection with Erasure Capability,” NIST White Paper, May 31, 2018.
18. V. Sivaraman, "High Throughput Cryptocurrency Routing in Payment Channel Networks," USENIX NSDI 2020.
19. ISO/IEC 17788:2014 “Information technology — Cloud computing — Overview and vocabulary”, https://www.iso.org/standard/60544.html

The Standards People

ETSI
06921 Sophia Antipolis CEDEX, France
Tel +33 4 92 94 42 00
info@etsi.org
www.etsi.org

This White Paper is issued for information only. It does not constitute an official or agreed position of ETSI, nor of its Members. The views expressed are entirely those of the author(s).

ETSI declines all responsibility for any errors and any loss or damage resulting from use of the contents of this White Paper.

ETSI also declines responsibility for any infringement of any third party's Intellectual Property Rights (IPR), but will be pleased to acknowledge any IPR and correct any infringement of which it is advised.