ETSI GS PDL 028 V1.1.1 (2025-06)

Permissioned Distributed Ledger (PDL);
Specification utilizing PDL to Standardized IoT Service Layer Platform oneM2M

Reference
DGS/PDL-0028_Study_SLP_oneM2M

Keywords
blockchain, IoT, 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

Contents

List of figures

List of tables

Intellectual Property Rights

Essential patents

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

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

Trademarks

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

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

Foreword

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

Modal verbs terminology

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

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

Executive summary

The present document investigates the integration of Permissioned Distributed Ledger (PDL) technology with the oneM2M IoT service layer platform to address security and data integrity challenges in centralized IoT systems. It examines how PDL's decentralized architecture can enhance data security, provide tamper-proof record-keeping, and implement robust access control mechanisms within IoT environments. The present document provides a detailed proof of concept demonstrating PDL-based access authentication for IoT systems, focusing on ticket validation and smart gate control. Through practical implementation examples and architectural analysis, it establishes the feasibility and benefits of standardized interworking between oneM2M and PDL platforms, offering insights for future development of secure, scalable IoT infrastructure with Blockchain integration.

Introduction

The present document examines the integration of Permissioned Distributed Ledger (PDL) technology with the oneM2M IoT service layer platform to address critical challenges in existing IoT infrastructures. Current centralized IoT platforms face significant vulnerabilities including data tampering, single points of failure, compromised device management, and inadequate access control mechanisms. These weaknesses become increasingly concerning as IoT deployments scale and manage more sensitive data across diverse applications.

Traditional oneM2M implementations typically rely on conventional database technologies that, while functional, cannot guarantee data immutability or provide decentralized verification. Meanwhile, standalone PDL solutions often lack standardized interfaces for seamless integration with established IoT ecosystems. The convergence of these technologies presents a compelling solution that leverages the complementary strengths of both platforms.

By integrating PDL's immutable ledger capabilities with oneM2M's comprehensive service layer functions, organizations can implement IoT systems with enhanced data integrity, transparent yet secure access control, resilience against network failures, and automated trusted execution through smart contracts. This integration addresses the fundamental security and reliability challenges that have hindered wider adoption of IoT solutions in mission-critical applications.

The present document is structured as follows: clause 1 defines the scope, while clauses 2 and 3 provide references and terminology. Clause 4 discusses Blockchain applications in IoT contexts, followed by introductions to oneM2M (clause 5) and PDL (clause 6) technologies. Clause 7 explores practical utilization of PDL within oneM2M systems, while clause 8 presents a proof of concept demonstrating the interworking capabilities. Finally, clauses 9 and 10 offer recommendations and conclusions for implementing this integrated approach.

1 Scope

1.1 In-scope

The present document covers the utilization of Permissioned Distributed Ledger (PDL) ETSI GS PDL 012 [1] technology for the oneM2M ETSI TS 118 101 [2] platform and includes a Proof of Concept (PoC) implementation. The objectives of the present document are to:

- Define the need for standardized interworking between oneM2M and PDL platforms
- Establish use cases for the interworking between oneM2M and PDL platforms
- Identify impacts and requirements for both oneM2M and PDL platforms to enable interworking
- Outline detailed procedures for the interworking between oneM2M and PDL platforms
- Demonstrate the feasibility of PDL-based access control integration with oneM2M
- Present a reference architecture for combining PDL and oneM2M technologies
- Analyse the security benefits of using distributed ledger technology with IoT services

The approach taken in the present document focuses on demonstrating the feasibility of standardized interworking between IoT (represented by oneM2M) and Blockchain (represented by PDL) platforms.

1.2 Out of scope

The following topics are considered outside the scope of the present document:

- Detailed implementation guidelines for specific PDL platforms beyond the conceptual framework
- Performance benchmarking or comparative analysis of different Blockchain technologies
- Comprehensive security analysis beyond the identified use cases
- Modifications to the core oneM2M standard specifications
- Business models and commercial deployment considerations
- Legacy system migration strategies
- Interoperability with non-PDL Blockchain implementations
- Protocol-level optimizations for IoT-Blockchain communications
- Implementation-specific details beyond what is necessary for the presented PoC

The present document serves as a study of integration possibilities rather than a normative specification for implementations.

2 References

2.1 Normative references

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

Referenced documents which are not found to be publicly available in the expected location might be found in the ETSI docbox.

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

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

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

[2] ETSI TS 118 101: "oneM2M; Functional Architecture (oneM2M TS-0001)".

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 may be useful in implementing an ETSI deliverable or add to the reader's understanding, but are not required for conformance to the present document.

[i.1] ETSI TR 118 501: "oneM2M Use Case collection".

[i.2] ETSI TR 118 525: "oneM2M; Application Developer Guide (oneM2M version 1.0.0 Release 1)".

[i.3] ETSI WP 48 PDL: "An Introduction of Permissioned Distributed Ledger (PDL)".

[i.4] Ethereum ERC-721: "Non-Fungible Token Standard".

3 Definition of terms, symbols and abbreviations

3.1 Terms

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

Application Dedicated Node (ADN): node that contains at least one Application Entity and does not contain a Common Services Entity

NOTE: See ETSI TS 118 101 [2].

Application Entity (AE): entity in the oneM2M architecture that handles application logic, providing specific M2M functionality

NOTE: See ETSI TS 118 101 [2].

Blockchain: distributed database technology that maintains a growing list of records (blocks) that are linked using cryptography

NOTE: Not attributed to a specific external source in the present document.

Common Services Entity (CSE): entity in the oneM2M architecture that delivers essential service functions like registration, discovery, and data management

NOTE: See ETSI TS 118 101 [2].

Common Services Function (CSF): functionality provided by a CSE, such as device management, registration, or security

NOTE: See ETSI TS 118 101 [2].

Decentralized Identifiers (DID): unique, self-sovereign identifiers stored in a PDL alongside network or service access credentials

NOTE: See W3C® standards (mentioned in clause 6.2).

ERC-721: standard for Non-Fungible Tokens on the Ethereum Blockchain

NOTE: See Ethereum ERC-721 [i.4].

European Blockchain Services Infrastructure (EBSI): Blockchain infrastructure that supports multiple storage strategies for digital identifiers

NOTE: European initiative, not attributed to a specific publication in the present document.

Infrastructure Node AE (IN-AE): AE that is registered with the CSE in the Infrastructure Node

Infrastructure Node CSE (IN-CSE): CSE which resides in the Infrastructure Node of the oneM2M architecture

NOTE: See ETSI TS 118 101 [2].

Internet of Things (IoT): network of physical objects embedded with sensors, software, and technologies to connect and exchange data with other devices over the Internet

NOTE: Not attributed to a specific external source in the present document.

Interworking Proxy application Entity (IPE): entity that connects different platforms, specifically oneM2M and PDL, facilitating bidirectional message exchange

NOTE: See ETSI TS 118 101 [2].

Machine to Machine (M2M): direct communication between devices using any communications channel

NOTE: Not attributed to a specific external source in the present document.

Middle Node AE (MN-AE): AE that is registered with the CSE in Middle Node

Middle Node CSE (MN-CSE): CSE which resides in the Middle Node of the oneM2M architecture

NOTE: See ETSI TS 118 101 [2].

Network Service Entity (NSE): entity in the oneM2M architecture that supplies fundamental network services to the CSE

NOTE: See ETSI TS 118 101 [2].

Non-Fungible Token (NFT): unique digital identifier recorded on a Blockchain, used in this context for ticket authentication

NOTE: See Ethereum ERC-721 [i.4].

oneM2M: global technical standard for interoperability across Machine-to-Machine and IoT platforms

NOTE: See ETSI TS 118 101 [2].

Permissioned Distributed Ledger (PDL): distributed ledger where participants require authorization to join and are governed by specific authorities

NOTE: See ETSI GS PDL 012 [1].

PDL Function (PDLF): function that connects mobile networks with PDL infrastructures

NOTE: See ETSI GS PDL 012 [1].

smart contract: self-executing contracts with the terms directly written into code, ensuring automatic execution of agreements on PDLs

NOTE: See ETSI GS PDL 012 [1].

Trusted Execution Environment (TEE): secure area of a processor that ensures data is protected with regards to confidentiality and integrity

NOTE: Not attributed to a specific external source in the present document.

Two-Factor Authentication (2FA): authentication method requiring users to provide two different types of verification before gaining access

NOTE: Industry standard security practice, not attributed to a specific external source in the present document.

3.2 Symbols

Void.

3.3 Abbreviations

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

2FA	Two-Factor Authentication
ADN	Application Dedicated Node
AE	Application Entity
AE-CSE	Application Entity/Common Services Entity
CSE	Common Services Entity
CSF	Common Services Function
DD	Data Demanders
DDM	Distributed Data Management
DIS	Discovery
DMR	Data Management and Repository
DP	Data Providers
DT	Data Trainers
DW	Data Warehouses
EBSI	European Blockchain Services Infrastructure
IN-AE	Infrastructure Node AE
IN-CSE	Infrastructure Node CSE
IoT	Internet of Things
IPE	Interworking Proxy application Entity
IRPs	Interface Reference Points
M2M	Machine to Machine
MN-AE	Middle Node AE
MN-CSE	Middle Node CSE
NFT	Non-Fungible Token
NSE	Network Service Entity
PDLF	PDL Function
PoC	Proof of Concept
SEC	Security
UE	User Equipment

4 IoT with Blockchain

4.1 Introduction

The integration of Blockchain technologies within the IoT service layer platform heralds a transformative era for various industries, significantly bolstering data integrity, system security, and device control. As centralized IoT platforms often rely on cloud-centric models, they face inherent risks such as data tampering, unauthorized access, and compromised device management due to their centralized structure. The decentralized nature of Blockchain, as a distributed ledger, presents a robust solution to these vulnerabilities.

By incorporating Blockchain into IoT, each data transaction is immutably recorded across multiple nodes, ensuring that system integrity remains intact even if a single point fails. This resilience, combined with the automation capabilities of smart contracts, can accelerate transaction processes and enhance overall system performance. The immutable record-keeping feature of Blockchain makes data falsification and unauthorized modifications impossible, while access to sensitive information is conditionally granted, ensuring that only verified users can engage with the system.

Thus, the synergy between Blockchain and IoT platforms is not just a step towards mitigating existing security concerns but a leap towards a more secure, efficient, and trustworthy digital infrastructure for IoT services. In the following clause 4.2, various use cases that show the synergetic benefits of integrating Blockchain with IoT platforms are described.

4.2 Use cases of using Blockchain in IoT

4.2.1 Enhancing Transparency and Confidentiality in the Wine Industry through Blockchain and IoT Technologies

In the wine industry, the integration of Blockchain with IoT technologies presents a compelling use case for ensuring the transparency and confidentiality of data from vineyard to consumer. IoT devices, managed by a IoT service layer platform, collect crucial quality metrics from the winery, such as soil condition and grape maturity, and monitor environmental conditions within the wine cellar to maintain the wine's integrity. These IoT devices also track the wine's journey, including any flights it may take, ensuring that the conditions remain optimal for wine preservation.

The data collected at each stage is securely stored on a private Blockchain network, guaranteeing the integrity of the information due to Blockchain's immutable nature. This data management approach allows wine applications to utilize general IoT services such as smart city data and access Blockchain-stored data, ensuring that the provenance and handling of the wine are transparent and meet regulatory compliance for data privacy.

The high-level use case depicted in Figure 1 herewith shows an IoT service layer platform with Blockchain employed for global service continuity, connecting private Blockchain networks from different regions through a public Blockchain network. This ensures a seamless and secure exchange of data across borders, where each bottle's journey - from the winery data collection, through storage in the wine cellar, the flight to international locations, and finally to the retail shop in Korea - is documented and verifiable.

By leveraging the IoT service layer platform, the wine industry can manage data effectively while utilizing Blockchain technology to ensure the data's integrity. This integration is pivotal for enhancing trust across the wine supply chain, providing stakeholders with a reliable record of the wine's quality and handling, and ultimately enhancing consumer confidence in the products they purchase.

The below shows several features to support this use case:

1) Data Collection through IoT: IoT sensors are deployed across the winery to collect a wide range of data. These sensors monitor the quality of the grapes, environmental conditions in the wine cellar, and the transportation of wine bottles to various outlets. The IoT service layer platform aggregates and manages this data, ensuring it is available for real-time monitoring and decision-making processes.

2) Data Integration with Blockchain: Selected data that require high integrity, such as quality certifications, origin of grapes, and cellar conditions, are recorded onto a Blockchain network. This could include timestamped entries for each batch of wine produced, ensuring a tamper-proof ledger. The Blockchain network provides an immutable record, guaranteeing the integrity of the wine-related data and enabling traceability throughout the wine supply chain.

3) Transparency and Confidentiality: Stakeholders in the wine value chain, including regulators, distributors, and consumers, can access the IoT platform to receive verified information about the wine's quality and journey. While the IoT platform provides general service data, the Blockchain network ensures that the data's integrity is maintained, and sensitive information remains confidential due to its encryption and permissioned access features.

4) Application and User Interaction: Wine applications can utilize the data managed by the IoT service layer platform for various services such as quality assurance, inventory management, and customer engagement. Customers can scan QR codes on wine bottles to view the Blockchain-stored data, such as harvest date, cellar conditions, and delivery details, thereby gaining confidence in the wine's authenticity and quality.

Benefits of this use case include:

- **Integrity:** Blockchain's immutable ledger means that once data about a wine batch is recorded, it cannot be altered, providing an unbreakable chain of custody.
- **Transparency:** Every stakeholder has access to up-to-date and accurate information about the wine production and distribution process.
- **Confidentiality:** Sensitive data is protected through Blockchain's secure and encrypted mechanisms, ensuring that only authorized parties can access it.
- **Quality Assurance:** The integration of IoT and Blockchain allows for the constant monitoring of production and storage conditions, directly correlating to the quality of the wine.

This use case demonstrates how Blockchain and IoT technologies can revolutionize the wine industry by providing a reliable and secure way to assure the quality and origin of each bottle, thereby enhancing consumer trust and regulatory compliance.

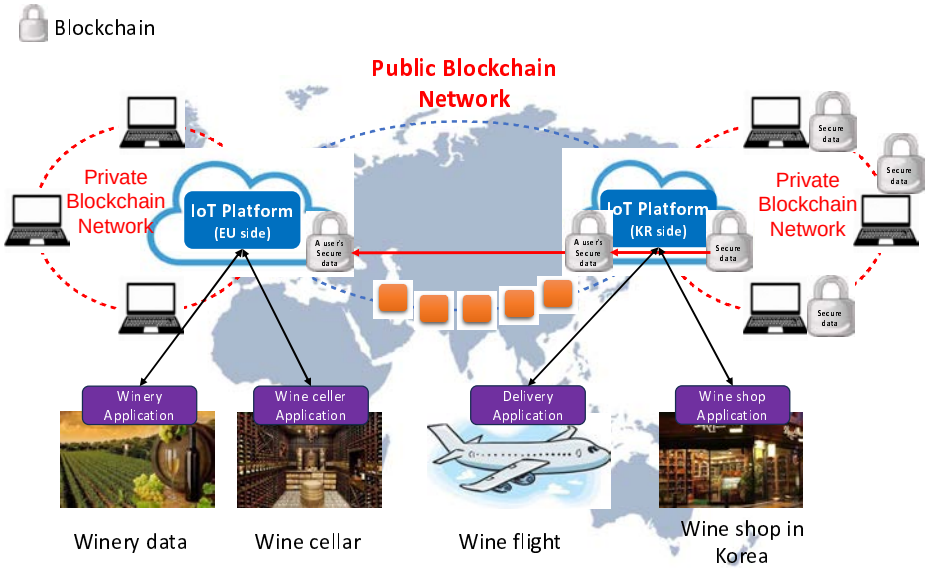


Figure 1: Using Blockchain and IoT technologies for the Wine Industry

5 Introduction to oneM2M

5.1 Overview of oneM2M

Founded in 2012, oneM2M is a collaborative project uniting eight prominent ICT standards development bodies globally, including ARIB and TTC from Japan, ATIS and TIA from the United States, CCSA from China, ETSI from Europe, TSDSI from India, and TTA from Korea. Its mission is to establish a universal technical standard that facilitates interoperability across Machine-to-Machine (M2M) and IoT platforms, encompassing everything from system architecture and API specifications to security protocols and registration processes. Through global consensus, oneM2M devises end-to-end M2M specifications, utilizing shared use cases and foundational architectural principles that span a multitude of M2M applications.

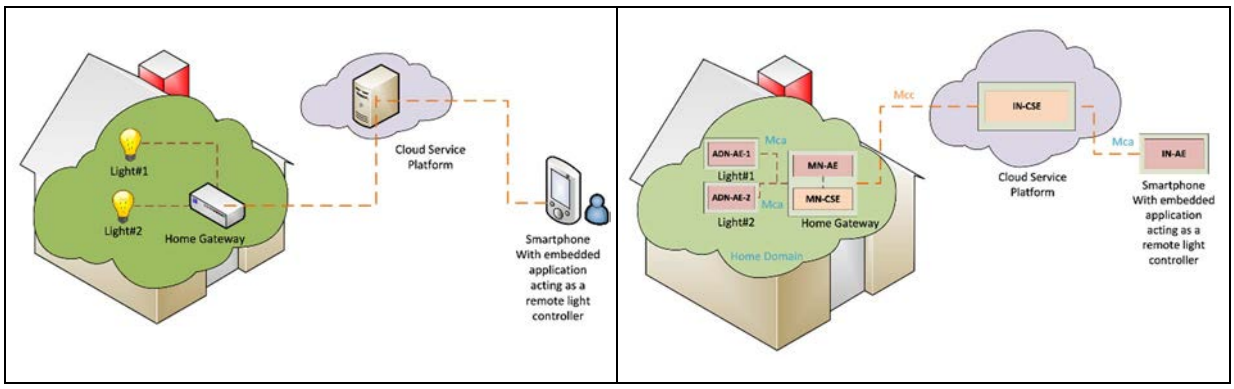
Acting much like a decentralized operating system for the IoT, oneM2M features a middleware service layer comprised of an array of Common Service Functions (CSFs). This middleware bridges the gap between applications and the underlying connectivity layers, with CSFs accessible via RESTful APIs for both applications and IoT devices. This service layer can be integrated within field devices, sensors, gateways, and backend cloud services, fostering collective intelligence within dispersed IoT ecosystems.

The oneM2M Service Layer Is essentially a software stratum Interfacing IoT applications with the hardware necessary for processing and connectivity, usually operating over IP, while also accommodating non-IP data transmission through interworking proxies.

This Service Layer supplies a suite of common service functions essential for IoT applications, with oneM2M's standards currently detailing fourteen such functions. These functions provide a modular toolkit for developers, who can start with essential services like device management, registration, and security, and gradually integrate more sophisticated capabilities for advanced applications, such as those requiring semantic interoperability and location-based services.

5.2 oneM2M use cases

OneM2M has been at the forefront of crafting use cases across a myriad of industry sectors, ranging from domestic environments like smart homes to industrial settings like smart factories. The use cases formulated are meticulously documented in ETSI TR 118 501 [i.1]. This particular clause draws upon a home lighting scenario where a user can remotely manipulate their home's lighting using their smartphone, all enabled through the oneM2M framework. The use case is mainly captured from ETSI TR 118 525 [i.2] that provides a use case for guiding application developers to develop applications using functionalities provided by a oneM2M service platform.



NOTE: It shows an overview of remote lights control use case (left) and oneM2M functional architecture of remote lights control use case (right).

Figure 2: oneM2M smart home use case

Figure 2 (left) provides a schematic of the home lighting use case, highlighting the primary elements involved:

1) The lighting fixtures are installed within a residence and are connected to a home gateway.

2) This gateway facilitates communication with a cloud-based service platform, which in turn allows for the lighting to be remotely operated via a smartphone.

3) The cloud platform is equipped with a suite of functionalities designed to streamline the control of home lighting through the smartphone. These functionalities encompass various services such as registration, discovery, data management, and group management, along with subscription and notification mechanisms.

4) The smartphone is equipped with an application that provides remote control over the home lighting system.

This app enables users to:

- Discover lighting fixtures within the home.
- Issue commands to toggle the lights on or off.
- Check the current state of the lights.

Figure 2 (right) delineates the alignment of the use case components with the corresponding entities within the oneM2M architecture.

The oneM2M architectural framework identifies two fundamental entity types: the Application Entity (AE) and the Common Services Entity (CSE). In our home lighting scenario, both the lighting fixtures and the smartphone are equipped with their respective AEs. A cloud-based Infrastructure Node CSE (IN-CSE) is maintained by the oneM2M Service Provider, while a Middle Node CSE (MN-CSE) is integrated within the Home Gateway. Interactions between an AE and a CSE are facilitated by the oneM2M-defined Mca reference point. Conversely, interactions between different CSEs utilize the Mcc reference point. In our scenario, the Light AEs communicate with the home gateway's MN-CSE, and the Smartphone AE communicates with the IN-CSE, both via the Mca reference point. Communication between the home gateway's MN-CSE and the IN-CSE occurs over the Mcc reference point.

To encapsulate, the applications in this use case are categorized as follows:

1) ADN-AE1: An embedded application within Light#1, capable of controlling the fixture and interacting with the MN-CSE through the Mca reference point.

2) ADN-AE2: A similar embedded application within Light#2, also capable of controlling the fixture and interfacing with the MN-CSE through the Mca reference point.

3) IN-AE: An application within the smartphone that directly interfaces with the oneM2M service platform's IN-CSE via the Mcc reference point, enabling remote control over Light#1 and Light#2.

4) MN-AE: An application within the home gateway that communicates with the MN-CSE through the Mca reference point, facilitating the overall connectivity and control within the home lighting system.

5.3 oneM2M architecture

The oneM2M reference architecture ETSI TS 118 101 [2], as illustrated in Figure 3, introduces multiple logical oneM2M entities such as the Application Entity (AE), Common Services Entity (CSE), and Network Service Entity (NSE), each fulfilling distinct roles across various layers. The AE handles the application logic, similar to how a smart home application operates on M2M services. The CSE delivers essential service functions like registration, discovery, and data management, accessible to both AEs and other CSEs. On the other hand, the NSE supplies fundamental network services to the CSE, including device activation, location services, and device management. These entities, located on devices ranging from mobiles to gateways or servers, communicate through designated oneM2M reference points, namely Mca for AE-CSE interactions, Mcc for CSE-CSE communications, and Mcn for CSE-NSE interactions. The Mcc reference point also facilitates communication between two CSEs within the Infrastructure Domain.

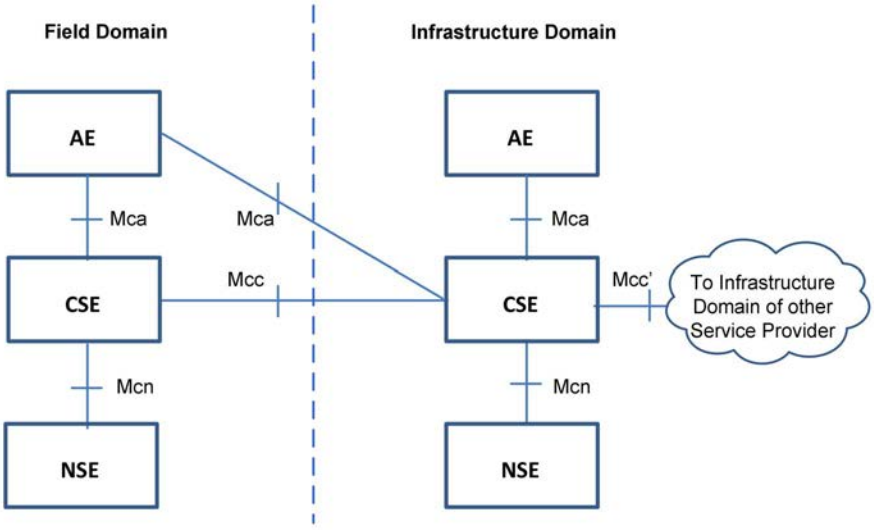


Figure 3: oneM2M function architecture showing logical entities and their reference points

oneM2M adopts a resource-oriented data model representing services as resources, with the CSEBase being the root for all resources within a CSE. This root can have various child resources like AEs, Containers, and accessControlPolicies. Similarly, AEs can host multiple child resources. oneM2M standards, which are widely utilized and evaluated across academia and industry, have been established across several releases.

Common Service Functions (CSFs) identified by oneM2M cater to a broad range of IoT domains, functioning like tools in a toolbox to address myriad IoT challenges. For instance, just as a screwdriver is versatile across automotive and aviation industries, oneM2M CSFs like device management or security apply across diverse IoT scenarios. In the initial phase of standardization, oneM2M members scrutinized numerous IoT use cases, deducing common requirements that led to the creation of CSFs. Moreover, oneM2M has standardized the execution of these functions by defining consistent APIs for access.

The non-domain-specific nature of CSFs allows for flexible application across various sectors, much like how general-purpose services of an operating system facilitate application functions such as file I/O. In this vein, oneM2M's Service Layer offers akin services to a multitude of IoT applications. CSFs are housed within a CSE, serving AEs via Mca and other CSEs through Mcc. The subsequent clause outlines oneM2M CSFs potentially intersecting with Blockchain technology:

- **Data Management:** The Data Management and Repository (DMR) CSF is tasked with data storage and transformation, supporting data aggregation, conversion, and storage for advanced analytics and semantic processing. This includes managing application data, user data, location, device details, and access permissions, laying the groundwork for Big Data.
- **Discovery:** The Discovery (DIS) CSF locates information about applications and services based on resource attributes, with the scope and outcome of discovery requests being contingent on filter criteria and access control policies.
- **Security:** The Security (SEC) CSF handles sensitive data protection, security administration, association establishment, and identity management, including access control measures that regulate operations on resources according to established policies and roles. It also ensures the protection of credentials and the implementation of security algorithms, while providing pseudonyms to safeguard entity identities.

6 Introduction to PDL

6.1 Overview of PDL

The ETSI Industry Specification Group (ISG) on Permissioned Distributed Ledger (PDL), established in 2018, focuses on laying down the groundwork for the operation of permissioned distributed ledgers. The ISG's mission is to create a standardized and open ecosystem of industrial solutions that can be adopted across various sectors, enhancing the application of these technologies and bolstering trust in information technologies underpinned by global, open telecommunications networks.

PDL is distinct from permissionless systems in that participants are not allowed to freely join or exit. Instead, they require authorization and are governed by specific authorities, which may be private entities or consortia. Governance and the associated access control policies are thus crucial in managing PDL systems.

PDLs offer selective visibility of information to the public, addressing privacy concerns more effectively than permissionless systems. They are also capable of employing more efficient consensus protocols, like proof-of-stake, to facilitate higher transaction speeds and improved energy efficiency. These features make PDLs an attractive choice for entities seeking the advantages of distributed ledger technologies, including decentralization and enhanced security, without the openness of permissionless systems.

Smart contracts on PDLs ensure automatic execution of agreements, benefiting from the inherent transparency and immutability of PDLs. Once set in motion, they function autonomously, eliminating the need for ongoing human oversight. PDLs transform Distributed Data Management (DDM) by bolstering trust, encouraging data integrity, and reinforcing data security. They enhance DDM applications, improving their roles as both data suppliers and users.

For situations where PDL nodes may be offline, resilience is preserved through strategies such as trusted configurations, standby proxies, ledger syncing, and comprehensive ledger management. Inter-ledger operability in PDLs facilitates record access across different ledgers. It can be unidirectional for selective data sharing or bidirectional for mutual data exchange and synchronization.

Key applications of PDLs include:

- **Mobile networks:** PDLs could facilitate spectrum trading and on-demand network access through a PDL Function (PDLF) that connects mobile networks with PDL infrastructures.

- **Data sharing:** PDLs ensure reliable data transactions, support secure computing environments, and apply smart contracts for secure, self-governing interactions.
- **AI:** PDLs confirm the authenticity of data and models, essential for distributed AI strategies, while addressing issues of trust, privacy, and motivation.

6.2 PDL use cases

One compelling use case showcasing the benefits of PDL technology is its ability to enhance mobile networks and services. Figure 4 shows a PDL Function in Mobile Core Networks. PDL technology offers significant improvements, including capabilities like spectrum trading and on-demand network access. The introduction of a new PDL Function (PDLF) could revolutionize the interface between mobile core networks and PDL networks. PDLF, an interworking function, is vital for interactions between PDL networks and participants, including end-device applications and mobile network control functions. It enables decentralized identification and authentication for User Equipment (UE), using Blockchain technology for verifying identities and service-specific credentials stored on a ledger. This method provides a more private and controlled digital identification and authentication process, facilitating seamless network access and service provision.

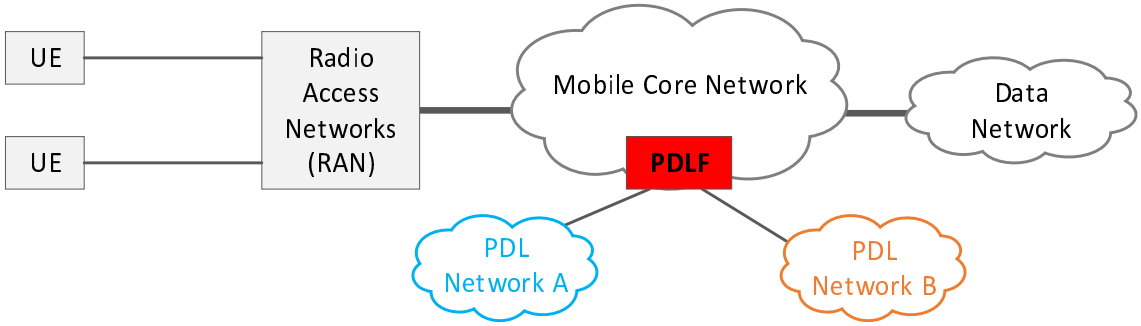


Figure 4: PDL Function in Mobile Core Networks

In current systems, devices needing initial network access are typically configured with default credentials by manufacturers, stored on a default credential server. Network operators, based on agreements with manufacturers, use these credentials for device onboarding. Post successful onboarding, devices receive actual network subscription credentials. PDL can change this process, especially for dynamic network access cases, by allowing real-time creation of digital identifiers and access credentials. These identifiers, like Decentralized Identifiers (DID) as per W3C standards, are unique, self-sovereign, and can be stored in a PDL alongside network or service access credentials. This approach, aligned with the eIDAS framework, enhances security and customer identification in digital services. The European Blockchain Services Infrastructure (EBSI) supports multiple storage strategies for these identifiers, catering to diverse legal requirements in use cases. PDL's application can extend to various scenarios, such as localized network services at events, or providing network access to users without subscription credentials.

According to the PDL whitepaper [i.3], PDL can solve the problem of sharing personal data for training. PDL act as a data carrier for training that requires multiple organizations to collect all data together. By doing so, the PDL ensures the reliability of training results, solves the problem of data silos, and enables centralized training in a trusted computing environment. It also realizes centralized training and maintains system security and autonomous interaction through smart contracts.

Figure 5 shows a trusted computing and data sharing use case based on PDL.

There are four actors in this use case:

- Data Demanders (DD): are responsible for purchasing the services and results provided by the data warehouse.
- Data Providers (DP): owners of the data resources or capabilities and are responsible for the availability, correctness, and accuracy of the data.
- Data Trainers (DT): the purchasers of data resources. Those can be developers or compilers of algorithms.
- Data Warehouses (DW): the platform where data transactions and operations are performed. It provides the necessary technology.

The Blockchain platform provides the necessary functions and services.

The following phases are defined in Figure 5:

1) Step 1 - DD, DP, DT, and DW register their workspaces on the Blockchain platform.

2) Step 2 - DP uploads the raw data, DW processes the data, generates a summary file, and registers it on the Blockchain.

3) Step 3 - the DD releases a specific demand to the data trainer, and the DT analyses the demand and requests the required data from the DP.

4) Step 4 - the DP uploads the requested data, and the DT uploads the algorithm to the TEE.

5) Step 5 - the DW summarizes the learning results and uploads them to the Blockchain, and returns the results to the DT.

6) Step 6 - the DT checks the veracity of the results, the DD pays the service fee, and the DT and DP are rewarded.

PDL can ensure security and process automation under the premise of data sharing. This allows PDL to solve the problem of sharing personal data for education and realize trusted computing and data sharing.

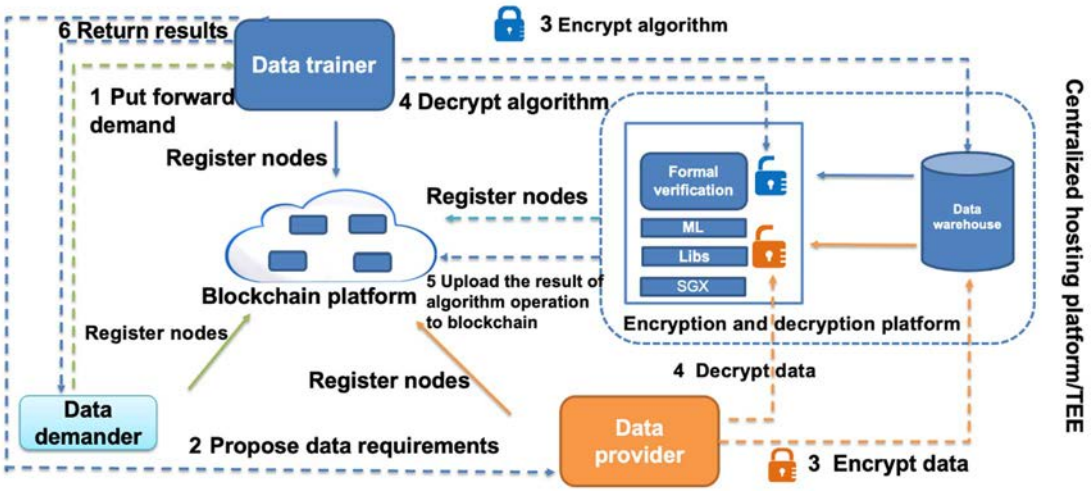


Figure 5: Trusted Computing and Data Sharing Scheme based on Blockchain [i.3]

6.3 PDL architecture

The PDL reference architecture [1], as depicted in Figure 6, offers a framework for permissioned distributed ledger systems. The PDL architecture introduces a layered approach to designing permissioned distributed ledger systems. It encompasses the User Layer, representing platform users the Applications Layer, hosting PDL-utilizing applications; and the Application Abstraction Layer, acting as an intermediary. The Applications Layer integrates applications utilizing PDL technology with core services. The Application Abstraction Layer acts as a mediator, enhancing the development of PDL structures. In the Platform Services Layer, necessary services for various applications are provided, split into several categories for flexibility and enhanced functionality. The Platform Services Layer offers a range of essential services, mainly Mandatory and Optional services. This layer ensures smoother development and interoperability.

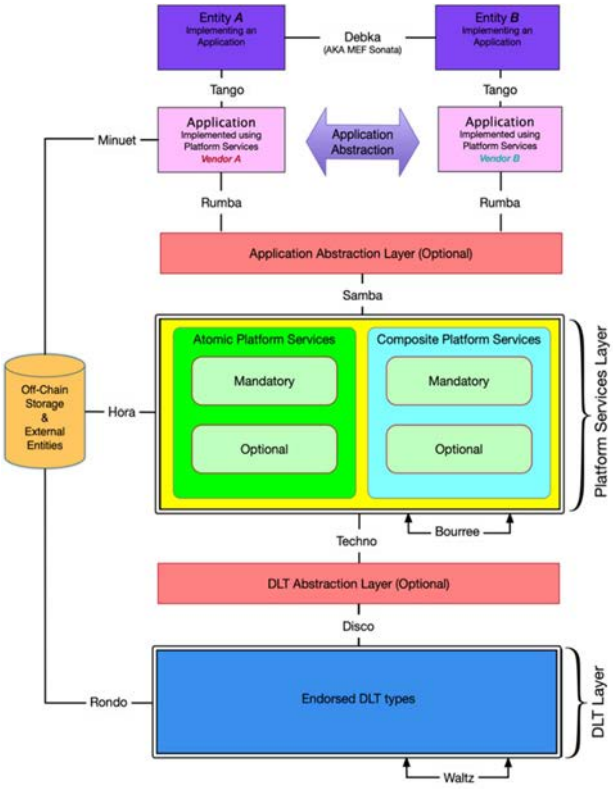


Figure 6: PDL Reference Architecture

Mandatory Services: These are services all compliant PDL platforms shall implement. They include:

- **Namespace Service:** Manages identifiers and namespaces for participants and data within the PDL network.
- **Consensus Service:** Enables agreement among participants on the validity of transactions and the state of the ledger.
- **Membership Service:** Manages participants' identities, access permissions, and roles within the network.
- **Cryptographic Service:** Provides cryptographic primitives for secure communication and data protection.
- **Data Storage Service:** Manages the storage, retrieval, and access control of data on the ledger.

Optional Services: These are services that may be implemented based on specific platform requirements or application needs. Examples include:

- **Smart Contract Service:** Enables execution of programmable logic on the PDL network.
- **Audit Service:** Records and provides access to historical data and events on the ledger.
- **Identity Management Service:** Provides advanced functionalities for managing participant identities and credentials.
- **Interoperability Service:** Enables communication and interaction with other distributed ledger technologies or systems.

The PDL Abstraction Layer ensures seamless interoperability among various PDL types. Endorsed PDL Types include specific PDL implementations adhering to ETSI ISG PDL standards. Lastly, Interface Reference Points (IRPs) facilitate internal and external communications, ensuring efficient interaction within the PDL ecosystem. This architecture aims to provide a flexible, scalable, and standardized framework for permissioned distributed ledgers, supporting a wide array of applications.

7 Utilization of PDL in oneM2M

7.1 Overview

This clause explores the use case of integrating PDL with oneM2M and examines the connection between these two standardized technologies. As the threat landscape for IoT evolves, securing the storage and processing of data within IoT service layer platforms becomes increasingly crucial. Traditional oneM2M platforms rely on conventional database technologies, which pose security risks due to their centralized nature, making them vulnerable to data tampering and unauthorized access. This motivates ongoing research into leveraging decentralized technologies for IoT platforms. Organizations like Hyperledger are actively exploring the use of Blockchain in IoT for secure data storage and processing, and the oneM2M initiative is also pursuing similar efforts. Given their inherent security and versatility, PDLs emerge as a promising solution for secure data storage and processing within the oneM2M platform.

The benefits of utilizing PDL technology for oneM2M platforms include the following:

- **Enhanced data integrity:** By utilizing the immutable record-keeping capabilities of Blockchain technology, data can be stored and processed securely against external data tampering threats.
- **Double-check access rights:** Can manage a secondary access rights policy by conditionally granting access to data in the PDL and access to data in the oneM2M platform. This allows sensitive information to be stored and processed more effectively and securely.
- **Single Point Of Failure (SPOF) recovery:** Even if a single point of oneM2M's devices fails, it can be effectively recovered because all relevant information is recorded in the ledger.
- **Decentralize IoT networks:** PDL can be used to effectively solve various problems caused by the existing centralized structure.

7.2 Two-factor authentication

Traditional access authentication systems rely on passes such as ID cards, magnetic swipe cards, or simple QR codes to grant entry to a secured area. These passes are presented to a scanner or reader, which verifies the information before allowing access. While functional, this approach has several security vulnerabilities:

- **Forgery and Duplication:** Physical passes can be forged, duplicated, or stolen, allowing unauthorized individuals to gain access.
- **Server Hacking:** If the authentication system's server is compromised, attackers can potentially grant access to unauthorized passes or create their own.
- **Simple Verification:** Traditional systems often only verify whether the pass presented is on a list of approved credentials, without deeper checks into the pass's authenticity.
- **Reduced Traceability:** Once a pass is scanned, traditional systems may not keep an immutable log of access, making it challenging to audit and trace back security breaches.
- **Single Point Of Failure (SPOF):** Centralized authentication servers represent a single point of failure, which could be exploited to disrupt the entire access control system.

A PDL-based access authentication system that uses distributed ledger for authentication can address these vulnerabilities by leveraging the security and immutability features of Blockchain technology. In highly secure environment like international airports, where advanced authentication systems are essential, IoT platforms utilizing Permissioned Distributed Ledger (PDL) can provide enhanced access control policies based on trust and privacy. Furthermore, implementing Two-Factor Authentication (2FA) on the oneM2M IoT platform, which adopts PDL's security framework, can facilitate safe and reliable operation of smart systems.

This clause outlines a use case scenario at international airports, depicted in Figure 7 herewith, where identity verification for security screenings is conducted through an IoT platform utilizing Blockchain-based authentication of passports and biometrics. Assuming that user biometrics and passport data are securely stored in the Permissioned Distributed Ledger (PDL) system, the process begins when users scan their fingerprint and passport at an IoT-enabled biometric smart gate. The scanned information is then transmitted to the IoT platform, which processes the data and performs two-factor authentication through the PDL system. Based on the authentication results from the PDL, the IoT system can then control access through the biometric smart gate.

Pre-condition Setup:

- **User Registration:** Users' biometric and passport information are securely uploaded to the Blockchain network. During this process, a unique transaction hash value is generated, encapsulating user data integrity.
- **Pass Generation:** This transaction hash value is then encrypted to further enhance security and stored in the Blockchain. An NFC tag or QR code is generated, containing this encrypted hash value, and distributed to the user. This code serves as the user's pass, which will be used for access authentication.
- **Administrator Access:** Only authorized administrators have the privilege to view and manage user information. They can add or revoke access credentials based on the security policies in place.

Authentication Flow:

1) Scan two-factor authentication data (Step 1): Users present their biometric (e.g. fingerprint) and passport information to an IoT-enabled access control device like a gate scanner at international airports.
2) Data acquisition & transmission (Step 2): The IoT device reads the two-factor information and sends it to the registered oneM2M IoT platform, where the data is stored in the corresponding resources.
3) Two-factors authentication (Steps 3 - 5): The application, which supports APIs for oneM2M and PDL, receives a notification of a newly scanned information. It then queries the PDL to authenticate the presented two-factor information, i.e. the fingerprint and passport ID. It compares the encrypted hash from the passport against the Blockchain records, using the transaction hash to retrieve and verify the corresponding information on the ledger.
4) Result communication (Step 6): After processing the 2FA query, the PDL informs the application whether the credentials are valid or not.
5) Access control command (Step 7): If the PDL confirms the result of 2FA as authentic, the application issues a command to the oneM2M platform, which in turn triggers the access control mechanism.
6) Entry and Record (Step 8): The door unlocks, granting access. Concurrently, the event - comprising the user's identity, timestamp, and access point details - is recorded on the Blockchain. This record is immutable and tamper-proof, assuring a reliable audit trail.

This system offers robust security by preventing unauthorized access through forged passes or compromised servers, as the immutable nature of the Blockchain ensures that only passes with a verifiable transaction hash grant access. It combines the distributed trust model of Blockchain with oneM2M's standard-based interoperability framework, offering a sophisticated access control system that is secure, reliable, and highly resistant to tampering.

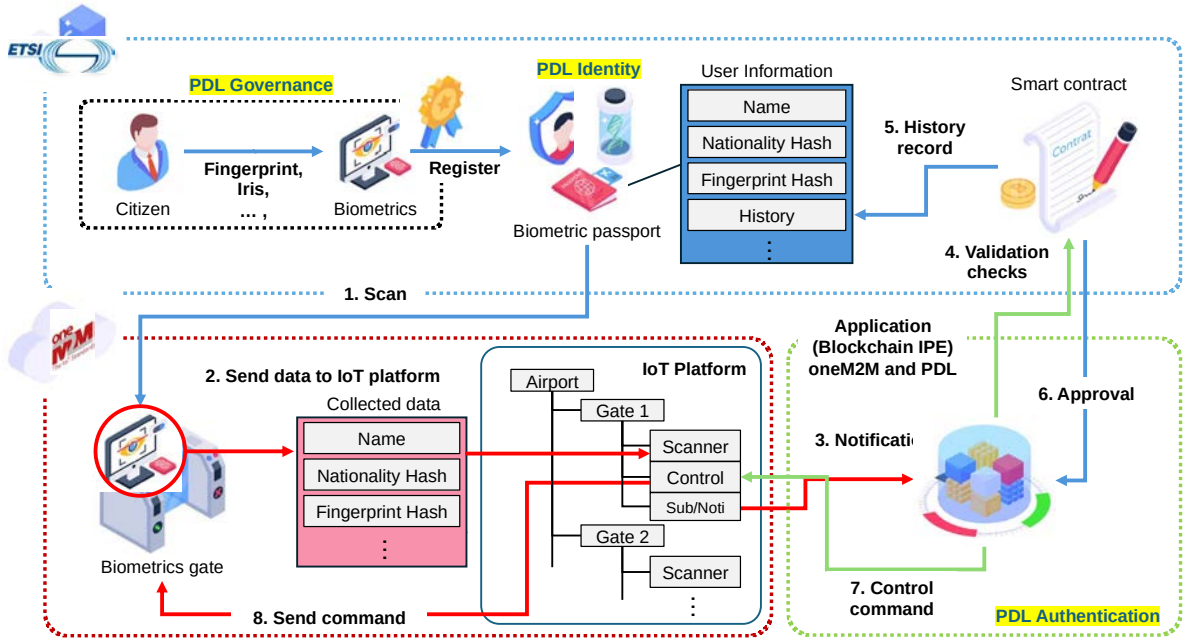


Figure 7: 2FA for biometric gates at an airport using oneM2M and PDL

7.3 Interworking between oneM2M and PDL

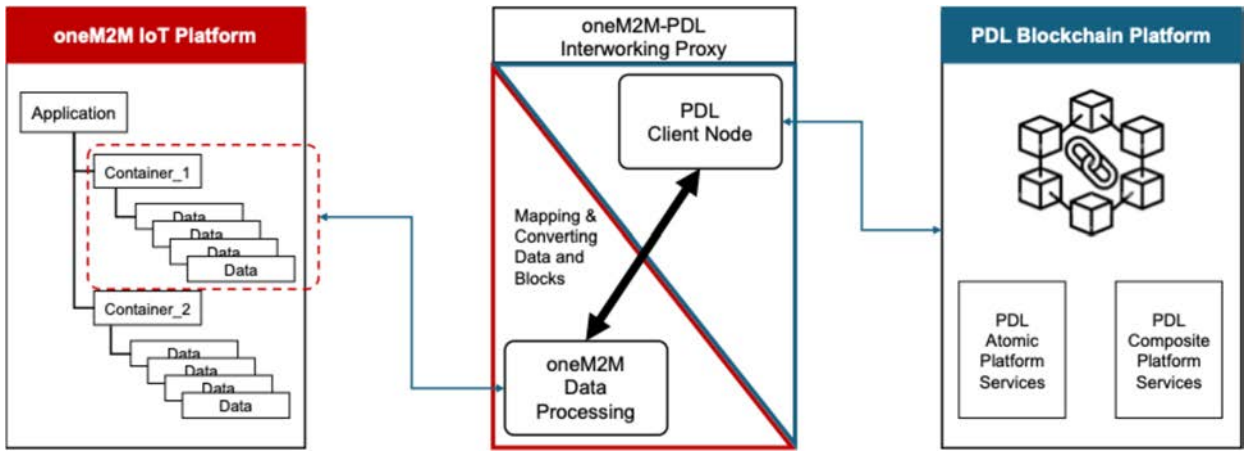


Figure 8: Interworking Proxy for oneM2M and PDL

The Interworking Proxy Entity (IPE) in the oneM2M Platform, depicted in Figure 8 above, can exist either internally or externally, and primarily monitors REST operations (i.e. CREATE, RETRIEVE, UPDATE, and DELETE) on data intended to be managed in Blockchain platforms. The IPE handles requests related to the REST operations of target data within the oneM2M platform. It maps these requests to the corresponding APIs of the PDL platform and performs necessary data transformations and mappings.

For instance, if data is created as a resource on the oneM2M platform and set to be managed via Blockchain, the IPE receives this request, processes the necessary information for storage on the PDL Blockchain platform, and requests the data be stored as a transaction on the PDL platform using predefined API mappings. Once stored, the IPE acquires an ID for accessing the transaction and adds this to the metadata of the corresponding resource stored on the oneM2M platform to facilitate future retrievals.

To support oneM2M-PDL interworking, container resources in the oneM2M platform can support additional labels such as:

- **Interworking Type:** PDL

- **Entity ID:** Target PDL platform ID
- **Target type:** Type of target PDL platform
- **Transaction ID:** Identifier of the corresponding transaction in the PDL platform

8 Possible Proof of Concepts (PoCs)

8.1 Overview and considerations

The purpose of the oneM2M-PDL interworking Proof of Concept (PoC) is to validate the practicability of data management support technology between IoT and Blockchain platforms through actual implementation. Specifically, it aims to demonstrate that data and device control, along with access control, are feasible by validating scanned NFT ticket information from an NFT scanner managed by the oneM2M IoT platform on the PDL Blockchain platform and accordingly controlling the smart gate registered on the oneM2M IoT platform through the IPE.

8.2 System design

Traditional access authentication systems face multiple security vulnerabilities, including forgery, tampering, and server hacking. For example, issues with scalpers who steal identities to buy and illegally transfer bulk tickets for cultural performances highlight the need for a system that ensures access only with legitimately issued tickets. This clause introduces a new ticket issuance and anti-counterfeiting solution using Blockchain's Non-Fungible Tokens (NFTs) to combat scalping and counterfeit tickets. The schematics of such solution are depicted in Figure 9 herewith. Integrated with an IoT access control system, it allows entry solely to individuals holding validly issued tickets. Utilizing a decentralized distributed ledger enhances security by preventing unauthorized access from forged passes or compromised servers. The system employs Blockchain standard PDL and IoT standard oneM2M technologies, enhancing interoperability and facilitating future expansion with other IoT devices.

A oneM2M access authentication system based on a Permissioned Distributed Ledger (PDL) can address many vulnerabilities by leveraging the security and immutability features of Blockchain technology. IoT platforms that utilize a PDL and NFT to prevent scalping - using macros to buy tickets in bulk and resell them at a profit - and to prevent ticket tampering can provide enhanced access control policies. Additionally, implementing two-Factor Authentication (2FA) on the oneM2M IoT platform, which adopts PDL's security framework, can effectively prevent scalping and ticket tampering, ensuring that access control systems operate securely and reliably.

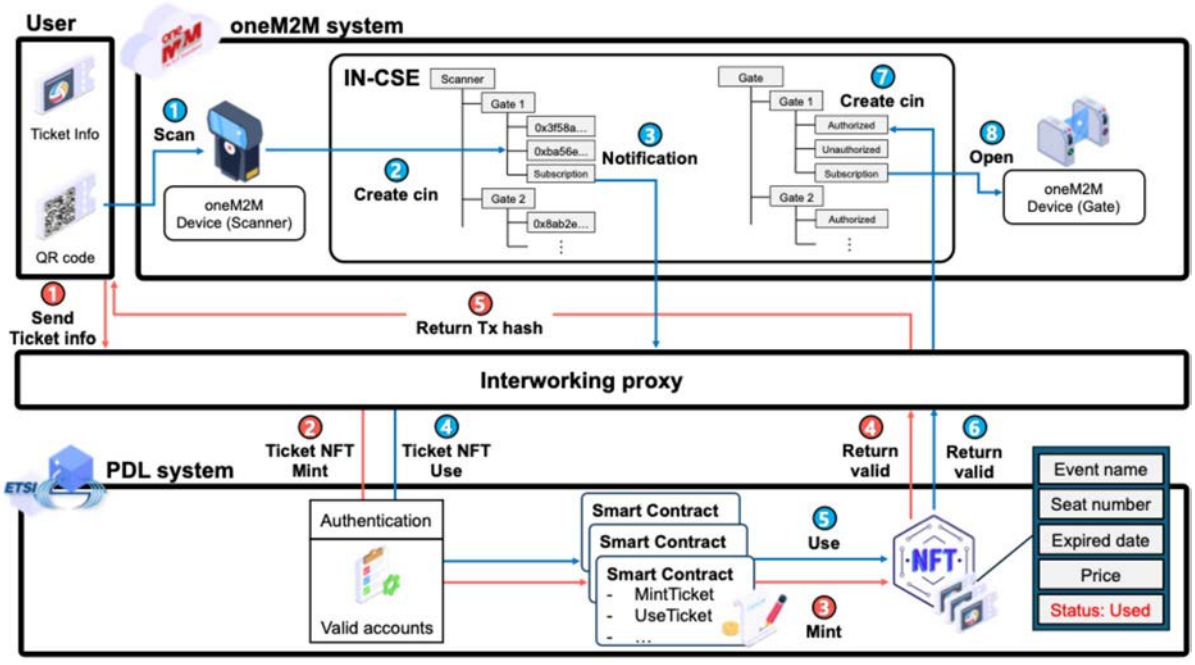


Figure 9: PDL-based oneM2M access authentication system overview

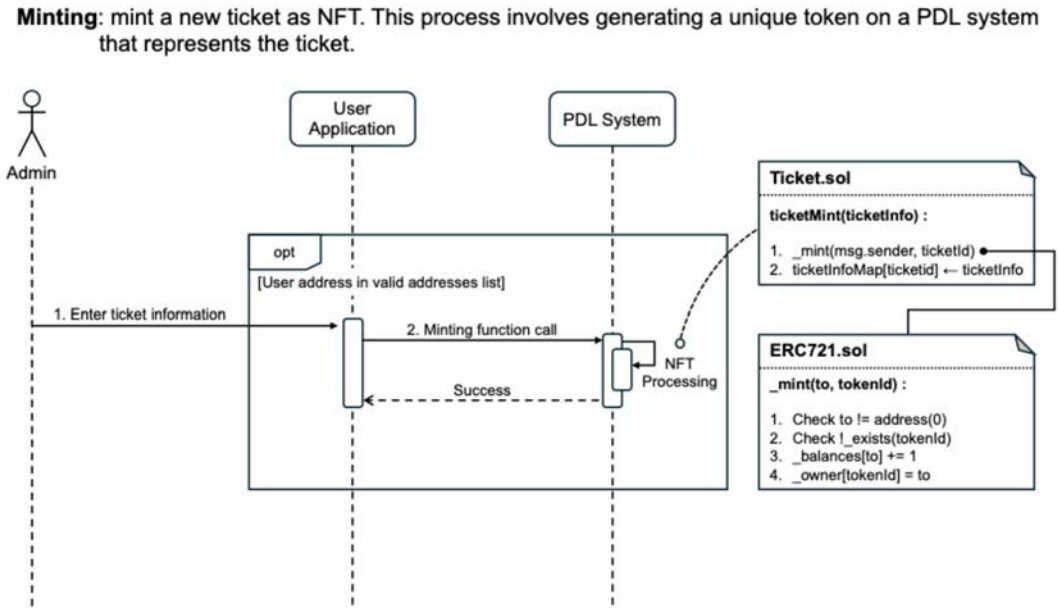


Figure 10: Ticket minting sequence diagram

Figure 10 shows the ticket NFT minting sequence diagram in the proposed system. The process involves generating a unique token on a PDL system that represents the ticket. After the minting process, validate tickets can be created with NFTs through the PDL system to access.

Ticket minting flow:

1) A user who exists in the list of valid addresses enters the ticket's information into the User application.

2) The user application calls the minting function of the PDL system.

3) The minting function creates an NFT ticket for the called user through a smart contract.

Figure 11 herewith shows a sequence diagram of the ticket listing and sales function. The procedure describes how to list the created NFTs for sale on a marketplace or ticket applications that support NFT transactions, allowing buyers to purchase or bid on the ticket. To prevent ticket scalping, the system does not allow NFT tickets to be set higher than the purchase price when reselling them.

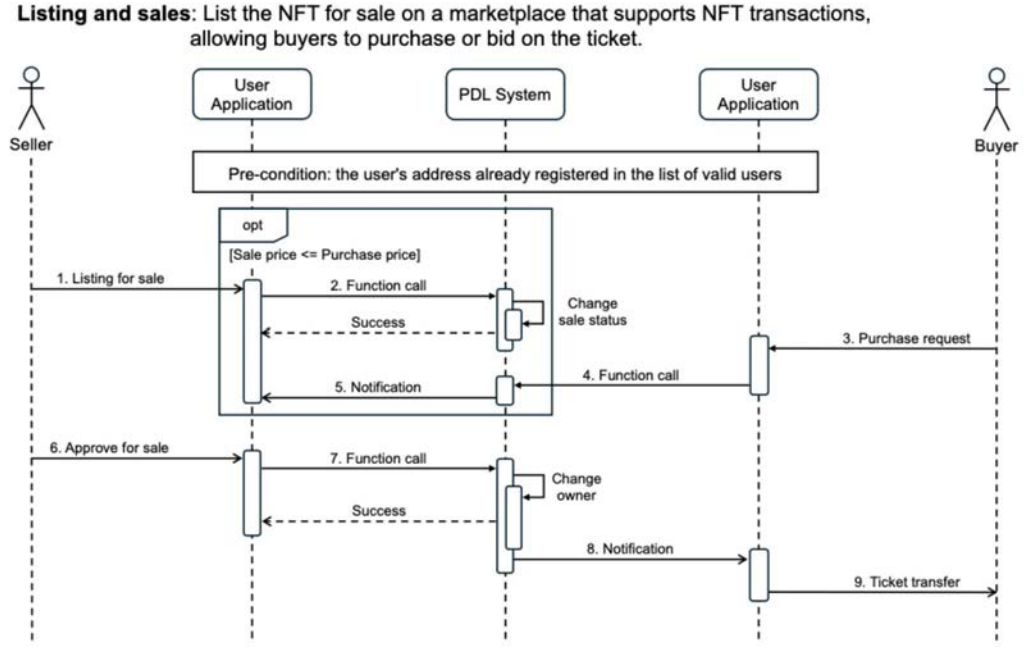


Figure 11: Ticket listing and sales sequence diagram to prevent scalping

Ticket listing and sales flow:

1) List a sale for less than the price that the seller purchased it for.
2) The user application calls the PDL system's register sale function, and the PDL system changes the NFT's status to sold.
3) The buyer checks the NFT ticket with the sold status and sends a purchase request to the User application.
4) The buyer's User application calls the PDL system's buy function.
5) The PDL system notifies the seller's User application that the buy function has been called.
6) The seller approves the sale.
7) The seller's User application calls the PDL system's approve sale function, and the PDL system changes the corresponding NFT ticket owner.
8) The PDL system notifies the buyer's User application that the purchase is confirmed and delivers the ticket.
9) The User application transfers the ticket to the buyer.

Figure 12 shows a sequence diagram of the PDL-based access authentication scheme in oneM2M. oneM2M can verify the validation of a ticket through PDL.

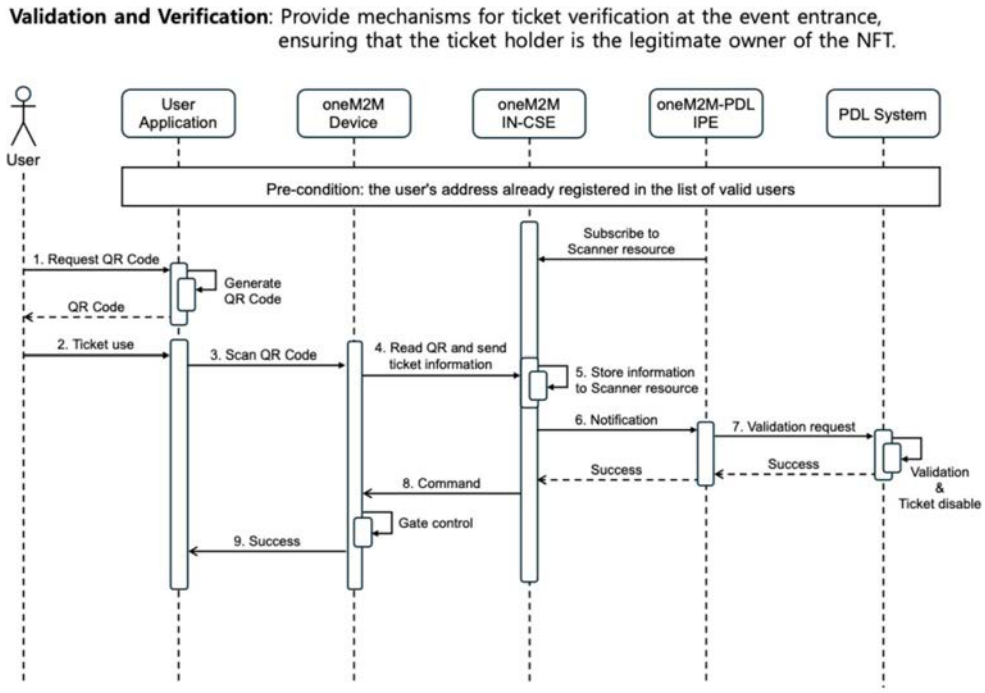


Figure 12: Ticket validation and verification sequence diagram for PDL-Based oneM2M access authentication

Ticket validation and verification flow:

1) When a user requests a QR code from the User application, the User application generates a QR code and delivers it to the user.
2) The user uses the ticket to scan the QR code in the User application with the scanner (oneM2M device).
3) The oneM2M device reads the QR code and passes the information to the oneM2M IN-CSE.
4) Store scanner resources in oneM2M IN-CSE.
5) When a scanner resource is saved in the oneM2M IN-CSE, it sends a notification to the oneM2M-PDL IPE that was subscribed to it.

6) The oneM2M-PDL IPE sends a validation request to the PDL system for that resource.

7) When the oneM2M-PDL IPE sends a validation request to the PDL system for that resource, the PDL system validates the NFT ticket and deactivates the ticket.

8) The oneM2M IN-CSE then sends a gate control command to the gate (oneM2M device) after verifying that the NFT ticket is valid.

9) The oneM2M device returns the success response to the User application.

8.3 Implementation details

8.3.1 Smart contract

The smart contract used in this PoC is designed to manage a decentralized ticketing system using Blockchain technology. It leverages ERC-721 [i.4] for Non-Fungible Tokens (NFTs) to ensure that each ticket is unique, traceable, and verifiable on the Blockchain. The contract allows tickets to be created, sold, transferred, and redeemed in a secure and transparent manner.

This smart contract provides a secure and efficient way to manage tickets on the Blockchain with emphasizing role separation and role-based access control. By defining specific roles and limiting their capabilities accordingly, the contract ensures that only authorized parties can perform sensitive tasks. This approach enhances security while ensuring the system is compliant with the features standardized in oneM2M and PDL, facilitating seamless integration and interoperability.

Key functionalities of the smart contract are as follows:

- **Ticket Minting:** Authorized users can create new tickets with specific attributes, such as event name, seat number, price, and expiry date. This ensures the integrity and scarcity of tickets by allowing only trusted entities to issue them.
- **Ticket Sale and Transfer:** Ticket owners can list their tickets for sale and set a selling price, while buyers can request to purchase these tickets. This feature enables a secure secondary market for tickets, allowing owners to resell their tickets to other buyers. The contract also prevents black market activities by ensuring that tickets cannot be sold for more than a pre-defined price set in the smart contract (e.g. the purchase price or no more than double the original purchased price).
- **Ticket Usage:** Ticket holders can redeem their tickets and mark them as used, preventing further transfer or resale. This helps prevent fraud by ensuring that tickets cannot be reused or resold once they have been used.
- **Role-based Access Control:** The contract uses role-based access control to manage permissions for different contract features. Sensitive tasks are restricted to authorized roles to enhance security, ensuring that only specified roles can perform specific actions.
- **Contract Pausing and Upgradability:** The contract can be paused in case of emergencies and supports upgrades for future enhancements. This provides flexibility and security by allowing the contracts to adapt to new requirements or cease operations if necessary.

Role separation and access control:

- The contract employs a robust role-based access control mechanism by defining specific roles, each with distinct permissions, to ensure that only authorized entities can interact with the smart contract's designated functions. In the proposed system, the following roles are defined:

  1) **Admin:** responsible for assigning and revoking roles and transferring ownership of contracts to other addresses. The admin oversees overall contract management and has full control over role assignments and administrative functions. Only admins can assign or revoke roles, providing centralized control over role assignments to prevents unauthorized access.

  2) **Minter:** Responsible for minting (creating) new tickets. Addresses with the minter role can invoke the "Create ticket" function for an event. This role restricts ticket creation to authorized addresses only, preventing unauthorized ticket creation that could lead to fraud.

3) **Pauser:** Has the ability to pause and resume (unpause) smart contract execution. This role allows contract operations to be halted in emergency situations to prevent unwanted activity. Only addresses with the pause role can execute these actions, enabling the system to respond immediately to security threats or technical issues.

4) **Upgrader:** Handles contract upgrades, with the authority to modify the contract code for improvements or corrections. Only addresses with the upgrader role can make such modifications, ensuring that only trusted entities can update the contracts, thus maintaining system integrity.

8.3.2 oneM2M-PDL IPE

The oneM2M-PDL Interworking Proxy Entity (IPE) acts as an intermediary that connects the oneM2M platform and the PDL platform, facilitating bidirectional message exchange. The IPE can communicate with the connected oneM2M platform using HTTP, CoAP, MQTT, or WebSocket, which are methods defined by the oneM2M standard. For communication with the PDL platform, it uses standard libraries provided by the target PDL platform to exchange messages.

The IPE promptly sends data received from oneM2M entities to the PDL, ensuring the integrity and reliability. The PDL uses smart contracts to automate the validation and execution of the data, returning the results to IPE. During this process, the IPE monitors the success of the smart contract in real time, controlling the opening and closing of the physical gate based on the results. In particular, the IPE achieves high throughput and low latency by employing asynchronous data processing and an event-based architecture, enabling it to maintain stable performance even in large-scale IoT environments.

With this comprehensive design and implementation, the oneM2M-PDL IPE playa a crucial role in building the oneM2M-PDL service by efficiently and reliably managing the collection, verification, execution, and control of IoT data. It is expected to contribute to the construction of various smart infrastructures such as smart cities and smart factories by integrating more functions and services in the future.

Table 1: oneM2M-PDL IPE API list

API name	Linked PDL smart contract functions	Parameter	Short description
/ticket/minter/remove	removeMinter	account	Revokes the MINTER_ROLE from a specified address, effectively removing their permission to mint new tickets. Only accounts with the DEFAULT_ADMIN_ROLE can execute this function.
/ticket/minter/add	addMinter	account	Grants the MINTER_ROLE to a specified address, allowing them to mint new tickets. This function can only be called by accounts holding the DEFAULT_ADMIN_ROLE.
/ticket/use	ticketUse	ticketId	Marks a specific ticket as used. Only the owner of the ticket can invoke this function. It ensures that the ticket is not expired and has not been used previously before updating its status to prevent reuse.
/ticket/sell	sellTicket	ticketId, price	Allows the owner of a ticket to list it for sale at a specified price. The function verifies that the ticket is valid, not expired, not already sold, and that the new sale price does not exceed the original price. Upon successful validation, it updates the ticket's sale status and price.
/ticket/mint	mintTicket	eventName, seatNumber, expiredDate, price	Creates (mints) a new ticket with the provided event details, seat number, expiration date, and price. Only addresses with the MINTER_ROLE can execute this function. It ensures that the ticket's expiration date is set in the future and initializes the ticket's status as available for sale and usable.
/ticket/has-role	hasRole	role, address	Checks if a given address possesses a specific role within the contract. Returns true if the address has the role, otherwise false. This function is inherited from the AccessControlUpgradeable contract and is used to manage role-based access control.

API name	Linked PDL smart contract functions	Parameter	Short description
/ticket/change-role	changeRole	role, oldAddress, newAddress	Transfers a specific role from one address (oldAddress) to another (newAddress). It first verifies that the oldAddress currently holds the role and that the newAddress does not already have it. Only accounts with the DEFAULT_ADMIN_ROLE can perform this role reassignment.
/ticket/buy-request	ticketRequestToBuy	ticketId	Allows a user to express interest in purchasing a specific ticket. It ensures that the ticket is listed for sale and that the requester is not the current owner. Upon successful request, it records the buyer's address in the ticketPurchaseRequests mapping for further processing.
/ticket/buy-confirm	ticketBuy	ticketId	Completes the purchase of a ticket that a user has previously requested. It verifies that the caller has indeed requested to buy the ticket and is approved to do so. Upon validation, it safely transfers ownership of the ticket to the buyer, updates the ticket's sale status, and removes the purchase request record.
/ticket/approve-sale	approveSale	ticketId	Approves a pending sale request for a specific ticket. Only the current owner of the ticket can call this function. It ensures that there is an active purchase request before approving the sale by setting the approved address, allowing the buyer to proceed with the purchase.
/ticket/info	getTicketInfo	ticketId	Retrieves detailed information about a specific ticket, including the event name, seat number, price, expiration date, sale status, and usage status. This function allows users to query the contract for the current state of a particular ticket.
/pdl-ipe/noti	ticketUse	transaction hash	Receive a json-formatted notification from oneM2M and run the ticketUse function with the corresponding ticketId via the transaction hash value.
/wallet	X	name, address, privateKey	It takes your name, address, and private key and saves the wallet address. The stored address will be used for future API calls.
/wallet/{name}	X	name	Returns information about the corresponding address based on the name.
/qr/generate	X	transaction hash	Generate a QR code with the transaction hash value.

9 Future Outlook and Recommendations

The integration of oneM2M and PDL platforms presents a promising path forward for securing IoT data and enhancing system interoperability. However, to fully realize the potential of this integration, several key areas require further attention and development.

First, a standardized interworking procedure between oneM2M and PDL platforms should be clearly specified. This will ensure seamless communication and data exchange between the two platforms, minimizing the risk of misalignment or data loss during transactions. The interworking proxy, which plays a crucial role in mapping data and coordinating interactions, needs to be refined to handle a variety of data types and operational contexts. This standardized procedure will facilitate broader adoption and consistent performance across different IoT applications.

Second, data mapping between the oneM2M service layer and PDL should be addressed in more detail. Since the two platforms may use different data structures and storage formats, creating a robust data mapping mechanism will be crucial for ensuring that information is accurately transferred and interpreted. This will also involve defining how specific types of metadata, such as timestamps and access control labels, are handled across the platforms to maintain data integrity and usability.

Finally, security aspects, including access control policies, need to be a core focus. As data flows between the oneM2M and PDL platforms, ensuring that only authorized entities can access sensitive information is critical. Implementing comprehensive role-based access control mechanisms and encryption standards will help safeguard data privacy and prevent unauthorized manipulation. Additionally, incorporating real-time monitoring of transactions and enforcing two-factor authentication can further enhance security, ensuring that both platforms maintain high standards of data protection in dynamic IoT environments.

By addressing these areas, the oneM2M-PDL interworking can become a foundational component for emerging IoT applications, offering enhanced security, transparency, and operational efficiency. This will ultimately contribute to the development of secure and scalable smart infrastructures, such as smart cities and factories.

10 Conclusion

The present document explored the feasibility of integrating Permissioned Distributed Ledger (PDL) technology with oneM2M, a standardized IoT service layer platform. The present document found that PDLs are a solution that can overcome the limitations of traditional centralized structures, enabling greater integrity, security, and access control of data. In particular, the tamper-proof nature of Blockchain plays a key role in ensuring the trustworthiness of IoT data management, and smart contracts can be used to automate data transactions and processing.

The integration of PDL and oneM2M can increase the reliability and security of data in IoT environments. Through the utilization of PDL, data can be securely managed in a decentralized manner and the integrity of the data can be maintained even in the event of a single point of failure. In addition, this study has shown that the proposed two-factor authentication method can compensate for the weaknesses of existing authentication systems and increase the security of access control.

Based on the study in the present document, it is necessary to define a standardized procedure for the interworking between PDL and oneM2M, as well as explore the possibility of its use in various IoT applications. Additionally, it is necessary to conduct further analysis of the oneM2M common service functions to identify the functionalities that require interworking with the PDL platform and proceed with standardization.

Annex A (informative): Change history

Date	Version	Information about changes
30 October 2023	V0.0.1	The initial baseline document is released.
26 June 2024	V0.0.2	Organized the structure of the baseline document (version 0.0.1) with the following structure: • clause 4: IoT with Blockchain • clause 5: Introduction to oneM2M • clause 6: Introduction to PDL • clause 7: Utilization of PDL in oneM2M • clause 8: Possible Proof of Concepts (PoC) • clause 9: Future Outlook and Recommendations Added an example of the use of Blockchain in IoT. Added content about oneM2M. Added content about PDLs. Added content about utilizing PDL in oneM2M.
05 September 2024	V0.0.3	Changed the structure of the baseline document (version 0.0.2) to the following structure: • clause 8.2: System design • clause 8.3: Implementation details Added content about Interworking between oneM2M and PDL. Added content about PoC overview and considerations. Added content on PoC system design.
03 October 2024	V0.0.4	Changed the structure of the baseline document (version 0.0.3) to the following structure: • clause 8.3: Implementation details • clause 8.3.1: Smart contract • clause 8.3.2: oneM2M-PDL IPE Added content about Implementation details. Added content about oneM2M-PDL IPE API List. Added reference [i.4].
23 October 2024	V0.0.5	Changed figure 9 Added content about conclusion.
9 January 2025	V0.0.6	In PDL#19 the document was changed from GR to GS.
28 March 2025	V0.0.7	Clean-up done by EditHelp.
10 April 2025	V0.0.8	Revised to polish the contents of Executive Summary and Introduction.
11 April 2025	V0.0.9	Final revision based on ISG Approval comments.

History

Document history		
V1.1.1	June 2025	Publication