International Telecommunication Union

# ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

**X.1408**
(10/2021)

SERIES X: DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

Secure applications and services (2) – Distributed ledger technology (DLT) security

# Security threats and requirements for data access and sharing based on the distributed ledger technology

Recommendation ITU-T X.1408

ITU-T X-SERIES RECOMMENDATIONS

**DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY**

| | |
|---|---|
| PUBLIC DATA NETWORKS | X.1–X.199 |
| OPEN SYSTEMS INTERCONNECTION | X.200–X.299 |
| INTERWORKING BETWEEN NETWORKS | X.300–X.399 |
| MESSAGE HANDLING SYSTEMS | X.400–X.499 |
| DIRECTORY | X.500–X.599 |
| OSI NETWORKING AND SYSTEM ASPECTS | X.600–X.699 |
| OSI MANAGEMENT | X.700–X.799 |
| SECURITY | X.800–X.849 |
| OSI APPLICATIONS | X.850–X.899 |
| OPEN DISTRIBUTED PROCESSING | X.900–X.999 |
| INFORMATION AND NETWORK SECURITY | |
| General security aspects | X.1000–X.1029 |
| Network security | X.1030–X.1049 |
| Security management | X.1050–X.1069 |
| Telebiometrics | X.1080–X.1099 |
| SECURE APPLICATIONS AND SERVICES (1) | |
| Multicast security | X.1100–X.1109 |
| Home network security | X.1110–X.1119 |
| Mobile security | X.1120–X.1139 |
| Web security (1) | X.1140–X.1149 |
| Application Security (1) | X.1150–X.1159 |
| Peer-to-peer security | X.1160–X.1169 |
| Networked ID security | X.1170–X.1179 |
| IPTV security | X.1180–X.1199 |
| CYBERSPACE SECURITY | |
| Cybersecurity | X.1200–X.1229 |
| Countering spam | X.1230–X.1249 |
| Identity management | X.1250–X.1279 |
| SECURE APPLICATIONS AND SERVICES (2) | |
| Emergency communications | X.1300–X.1309 |
| Ubiquitous sensor network security | X.1310–X.1319 |
| Smart grid security | X.1330–X.1339 |
| Certified mail | X.1340–X.1349 |
| Internet of things (IoT) security | X.1350–X.1369 |
| Intelligent transportation system (ITS) security | X.1370–X.1399 |
| **Distributed ledger technology (DLT) security** | **X.1400–X.1429** |
| Application Security (2) | X.1450–X.1459 |
| Web security (2) | X.1470–X.1489 |
| CYBERSECURITY INFORMATION EXCHANGE | |
| Overview of cybersecurity | X.1500–X.1519 |
| Vulnerability/state exchange | X.1520–X.1539 |
| Event/incident/heuristics exchange | X.1540–X.1549 |
| Exchange of policies | X.1550–X.1559 |
| Heuristics and information request | X.1560–X.1569 |
| Identification and discovery | X.1570–X.1579 |
| Assured exchange | X.1580–X.1589 |
| Cyber Defence | X.1590–X.1599 |
| CLOUD COMPUTING SECURITY | |
| Overview of cloud computing security | X.1600–X.1601 |
| Cloud computing security design | X.1602–X.1639 |
| Cloud computing security best practices and guidelines | X.1640–X.1659 |
| Cloud computing security implementation | X.1660–X.1679 |
| Other cloud computing security | X.1680–X.1699 |
| QUANTUM COMMUNICATION | |
| Terminologies | X.1700–X.1701 |
| Quantum random number generator | X.1702–X.1709 |
| Framework of QKDN security | X.1710–X.1711 |
| Security design for QKDN | X.1712–X.1719 |
| Security techniques for QKDN | X.1720–X.1729 |
| DATA SECURITY | |
| Big Data Security | X.1750–X.1759 |
| Data protection | X.1770–X.1789 |
| IMT-2020 SECURITY | X.1800–X.1819 |

*For further details, please refer to the list of ITU-T Recommendations.*

# Recommendation ITU-T X.1408

## Security threats and requirements for data access and sharing based on the distributed ledger technology

## Summary

A distributed ledger technology (DLT) is defined as a shared digital ledger, or a continually updated list of all transactions.

Data is accessed by a data controller (organization) and is possibly transferred to a data processor (organization) that will be responsible for processing the data on behalf of the data controller. A data controller should determine the purpose for which and the manner in which the data will be processed according to the constraints imposed by the data usage policy set by organizations.

In this context, there is a necessity for a trusted and transparent solution to enhance:

1) traceability of the data being accessed by data controllers and data processors directly or indirectly;

2) verifiability that if the data was accessed, used, and transferred without violating the data policy set by organizations; and,

3) changeability of data status in case of modification of data policy or any other cases.

An important aspect of this solution is to enable trust and transparency on accountability of data processing e.g., data provenance and usage tracking. It should offer transparent and controlled access, sharing and processing of data, so that unauthorized users or untrusted servers cannot process data without the authorization.

Recommendation ITU-T X.1408 focuses on the solution which is suitable for implementation using private-chain distributed ledger technology where data is accessed and shared less frequently. It specifies security requirements to improve traceability of data, verifiability of data, and changeability of data status.

Recommendation ITU-T X.1408 also specifies a reference model to describe data access and sharing based on the distributed ledger technology (DLT). It identifies entities and their roles and security threats for data access and sharing based on DLT. In addition, security requirements are specified to address these identified security threats.

## History

| Edition | Recommendation | Approval | Study Group | Unique ID* |
|---|---|---|---|---|
| 1.0 | ITU-T X.1408 | 2021-10-29 | 17 | 11.1002/1000/14801 |

## Keywords

Accountability, data access, data controller, data processor, distributed ledger technology, traceability, transparent.

[*] To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

## FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

## NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

## INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents/software copyrights, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the appropriate ITU-T databases available via the ITU-T website at http://www.itu.int/ITU-T/ipr/.

## Table of Contents

# Recommendation ITU-T X.1408

# Security threats and requirements for data access and sharing based on the distributed ledger technology

## 1 Scope

This Recommendation specifies a reference model to describe data access and sharing based on the distributed ledger technology (DLT). It identifies entities and their roles and security threats for data access and sharing based on DLT. In addition, security requirements are specified to address these identified security threats.

Issues related to privacy are out of scope in this Recommendation.

## 2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

None.

## 3 Definitions

### 3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

**3.1.1 control** [b-ISO/IEC 27000]: Measure that is modifying risk.

NOTE 1 – Controls include any process, policy, device, practice, or other actions which modify risk.

NOTE 2 – Controls may not always exert the intended or assumed modifying effect.

**3.1.2 data owner** [b-ISO/TR 14292]: Person having responsibility and authority for the data.

**3.1.3 de-identification technique** [b-ISO/IEC 20889]: Method for transforming a dataset with the objective of reducing the extent to which information is able to be associated with individual data principals.

**3.1.4 distributed ledger technology (DLT)** [b-ITU-T X.1400]: Technology that enables the operation and use of distributed ledgers.

**3.1.5 identity** [b-ISO/IEC 29100]: Set of attributes which make it possible to identify the personally identifiable information principal.

**3.1.6 off-chain** [b-ITU-T X.1400]: Related to a blockchain system, but located, performed or run outside that blockchain system.

**3.1.7 on-chain** [b-ITU-T X.1400]: Located, performed or run inside a blockchain system.

**3.1.8 pseudonym** [b-ISO/IEC 20889]: Unique identifier created for a data principal to replace the commonly used identifier [or identifiers] for that data principal.

NOTE – A pseudonym is sometimes also known as an alias.

**3.1.9 pseudonymization** [b-ISO/IEC 20889]: De-identification technique that replaces an identifier (or identifiers) for a data principal with a pseudonym in order to hide the identity of that data principal.

**3.1.10 sidechain** [b-ITU-T X.1400]: A blockchain system that interoperates with a separate associated blockchain system to perform a specific function in relation to the associated blockchain system.

NOTE – By convention the original chain is normally referred to as the "main-chain"(see clause 3.2.9), while any additional blockchains (see clause 6.8 in [b-ITU-T X.1400]) which allow DLT users (see clause I.3 in [b-ITU-T X.1400]) to transact on the main chain are referred to as "sidechains".

**3.1.11 smart contract** [b-ITU-T X.1400]: A program written on the distributed ledger system which encodes the rules for specific types of distributed ledger system transactions in a way that can be validated, and triggered by specific conditions.

**3.1.12 threat** [b-ISO/IEC 27000]: Potential cause of an unwanted incident, which may result in harm to a system or organization.

### 3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

**3.2.1 data controller**: Privacy stakeholder(s) that determines the purposes and means for processing data other than natural persons who use data for personal purposes.

NOTE 1 – A data controller sometimes instructs others (e.g., data processors, see clause 3.2.4) to process data on its behalf while the responsibility for the processing remains with the data controller.

NOTE 2 – This definition is adapted from PII controller in [b-ISO/IEC 29100].

**3.2.2 data owner agent**: A component that processes data on behalf of a data owner (see clause 3.1.2).

**3.2.3 data principal**: Natural person to whom the data relates.

NOTE 1 – Depending on the jurisdiction and the particular data protection and privacy legislation, the synonym "data subject" can also be used instead of this term "data principal".

NOTE 2 – This definition is adapted from PII principal in [b-ISO/IEC 29100].

**3.2.4 data processor**: Privacy stakeholder that processes data on behalf of and in accordance with the instructions of a data controller (see clause 3.2.1).

NOTE – This definition is adapted from PII processor in [b-ISO/IEC 29100].

**3.2.5 data usage contract**: A policy-based smart contract that specifies constraints on the usage and redistribution of any data obtained explicitly or implicitly by the data controller (see clause 3.2.1).

**3.2.6 explicit data**: Any data provided directly through interactions with the data owner (see clause 3.1.2).

**3.2.7 hybrid distributed ledger system**: Distributed ledger system that combines the privacy benefits of a permissioned distributed ledger system with the security and transparency benefits of a permissionless distributed ledger system.

**3.2.8 implicit data**: Any data acquired automatically by, for example, sensors of Internet of things (IoT) devices, applications installed in mobile devices, server, or database.

**3.2.9 main-chain**: Related to a distributed ledger system (DLT) system, but located, performed or run within that DLT system.

## 4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

| | |
|---|---|
| DLT | Distributed Ledger Technology |
| DoS | Denial of Service |
| IoT | Internet of Things |
| IPsec | Internet Protocol security |
| PII | Personally Identifiable Information |
| TLS | Transport Level Security |

## 5 Conventions

In this Recommendation:

The keywords "is required to" indicate a requirement which must be strictly followed and from which no deviation is permitted, if conformance to this Recommendation is to be claimed.

The keywords "is recommended" indicate a requirement which is recommended but which is not absolutely required. Thus, this requirement need not be present to claim conformance.

The keywords "is prohibited from" indicate a requirement which must be strictly followed and from which no deviation is permitted, if conformance to this Recommendation is to be claimed.

The keywords "can optionally" indicate an optional requirement which is permissible, without implying any sense of being recommended. This term is not intended to imply that the vendor's implementation must provide the option, and the feature can be optionally enabled by the network operator/service provider. Rather, it means the vendor may optionally provide the feature and still claim conformance with the specification.

## 6 Personally identifiable information (PII) protection in DLT

### 6.1 General PII protection principle

In this Recommendation, where data may contain PII elements PII protection should ensure compliance with the privacy principles of [b-ISO/IEC 29100], i.e.:

1) Consent and choice
2) Purpose legitimacy and specification
3) Collection limitation
4) Data minimization
5) Use, retention and disclosure limitation
6) Accuracy and quality
7) Openness, transparency and notice
8) Individual participation and notice
9) Accountability
10) Information security
11) Privacy compliance

### 6.2 Classification of DLT

There are three types of distributed ledger technology (DLT) systems: permissionless, permissioned and hybrid.

Permissionless distributed ledger systems are open to anyone validating blocks, without needing permission from any authority. In permissionless distributed ledger systems, it is not required for users to obtain permissions to use or operate the system. Theirs systems are often implemented using open source software, freely available to anyone who wishes to download them.

Permissioned distributed ledger systems are systems in which permissions are required. In permissioned distributed ledger systems, users validating blocks must be authorized. Since only authorized nodes are maintaining the distributed ledger, it is possible to restrict read access and to restrict who can issue transactions.

Hybrid distributed ledger systems combine the privacy benefits of a permissioned distributed ledger system with the security and transparency benefits of a permissionless distributed ledger system. This gives businesses significant flexibility to choose what data they want to make public and transparent and what data they want to keep private.

### 6.3 PII storage within distributed ledger

In theory, personally identifiable information (PII) can be stored within distributed ledgers, known as PII on-chain storage, although on-chain storage is strongly discouraged. If PII stored in the distributed ledger should be modified, deleted, updated or changed in some way, then possible options may include a hard fork in the chain or the cessation of the chain itself [b-ISO/TR 23244].

As the distributed ledger increases in size and as transactions are added, the accumulated data within the distributed ledger itself and links to external databases and storage may lead to negative effects, leading to the direct or indirect identification of a data principal. In addition, advanced analysis and profiling capabilities can also lead to negative or other effects, again leading to the direct or indirect identification of a PII principal.

An intermediary solution holds that when symmetric encryption is mandated by the system's design to store data on the distributed ledger, 256-bit keys are to be used as a minimum for systems that are meant to last for more than 10 years [b-FG DLT D4.1]. As the on-chain data can be accessed by anyone who has permission for the ledger, this case introduces a risk of data breaches stored in the ledger exposed to threats such as loss of keys, brute force attack, cryptanalysis attack.

### 6.4 PII storage outside distributed ledger

Where data is stored in PII off-chain storage outside distributed ledger, PII protection can be addressed by adopting the principles of ISO/IEC 29100.

DLT systems typically use hashes of data containing PII to allow it to be stored off-chain whilst a record of the data, confirming the existence of the data at a certain moment in time and its provenance and authenticity and enabling verification of its integrity, is held on the distributed ledger.

This facilitates data to be held off-chain whilst the integrity of the data referenced is maintained through use of the hashing function on the data. Identifiers can be used to point to data containing PII held off the chain, where these identifiers are not derived from the PII data itself and will probably only have a one-way relationship.

This "PII off-chain" storage provides strong protection of the transaction data. It may restrict access to the transaction data to authorized entities only, supporting use cases where participants may wish to keep this data of their multiparty transactions private from other DLT participants.

However, storing information "off-chain" negates many of the advantages of using a DLT in the first place. Although the use of hashes may highlight breaks, without transaction details, the DLT may no longer be a single, shared "source of truth." The issuance and trading of negotiable, fungible digital assets is no longer possible without reference to an on-chain position-keeping system.

Additionally, storing transaction data off-chain typically requires that both counterparties maintain their own record, or delegate that responsibility to a trusted third party, which brings with it the same costs and disadvantages as restricting read access to the distributed ledger.

A major challenge for DLT systems is the ability to ensure that a data or a block on a node and any associated off-chain storage has been deleted. DLT may also integrate with distributed file systems. These file systems also have the same challenge of ensuring that if a file is deleted confirmation is received that on each node the file was deleted.

## 7 Reference model for data access and sharing based on DLT

### 7.1 Overview

This clause describes an overview of data access and sharing based on DLT [b-Neisse], [b-FG DLT I-038]. A basic reference model for data access and sharing based on DLT is shown in Figure 1.

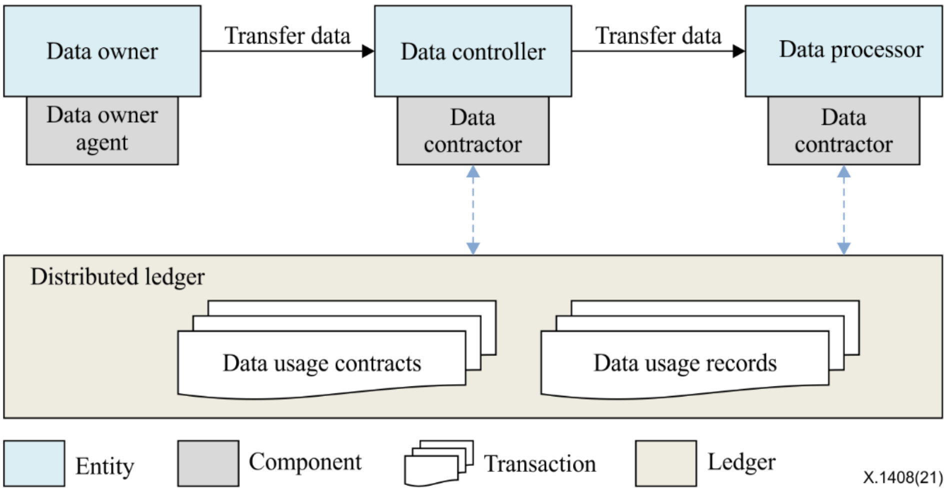

**Figure 1 – Reference model for data access and sharing based on DLT**

In this architecture, there are three main entities: the data owner, the data controller, and the data processor. In addition, data owner agent, data controller owner, and data processor agent act as data owner, data controller and data process, respectively. The data usage contract is a smart contract which is created by data owner. The smart contract includes the list of data hashes accessed by data controller and data processors, and the encoded usage control policy specifying the PII preferences of the data owner stating how his/her data is allowed to be used.

The functional model for data access and sharing based on DLT is shown in Figure 2.

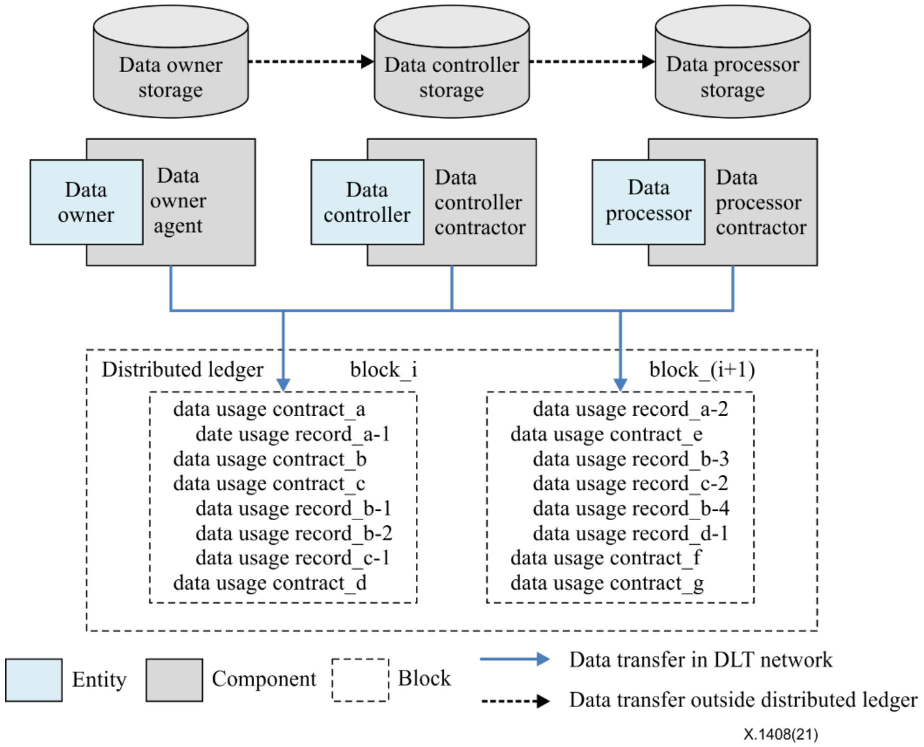

**Figure 2 – A functional model for data access and sharing based on DLT**

When the data owner subscribes to a data controller that acts as typically a service provider, it creates a policy-based data usage contract specifying constraints on the usage and redistribution of any data obtained explicitly or implicitly by the data controller. Data processor is a third-party organization that desires to get an access to data of the data owner. It requests the access to data of the data owner when it is authorized according to the redistribution policy in the data usage contract.

Data agent for each entity is an application to process data from data owners and to join the DLT network as a representative of each entity. Data agent fundamentally has two functions: 1) a component of a DLT system which creates a distributed-ledger-based data usage contract, executes data usage contracts as requests from data controllers or data processors, and generates transactions as it executes data usage contracts, 2) an off-chain storage which encrypts and stores collected data from the data owner securely, sends data align with verifications by the data usage contract, and establishes a secure channel for transmission.

Based on the functional model in Figure 2, the implementation of users sharing their own data based on DLT with third parties is shown in Annex B.

There are two types of data related to data owner: explicit data that is any data provided directly through interactions with the data owner such as documents, identifiers, attributes, any types of electronic files; implicit data that is any data acquired later without the data owner knowing during the provision of the service, for example, sensor data from IoT devices in the environment surrounding the data owner, data acquired by applications installed in mobile devices, or even server log files registering details of the network interactions between the data owner and data controller services (e.g., IP addresses).

The data usage contract in this model is a smart contract deployed in distributed ledger, acting as a data usage tracker, policy evaluation, and event logger that allow the data owner to easily check all data transfers and usage transactions providing assurance that only transactions conforming to the contract policies are authorized and registered in the distributed ledgers. It includes the list of hashed values of data that are accessed by data controller and data processors, and the encoded usage control policy specifying the data preferences of the data owners stating how his/her data is allowed to be used.

The data usage contract stores a list of the data of a data owner, which is accessible by data controller or data processor and is transferred to the data controller, including data instantiation and instance values, which are not stored in plaintext since the data usage contract could be deployed in a distributed ledger. Examples of data types are 'string', 'integer', or 'data', examples of data instantiations are 'name: string', 'e-mail: string', or 'date of birth: date', and examples of data instantiation value are name: 'User full name', and e-mail: 'user@host.com'.

It also contains the data usage contract policies that allow the data owner to easily check all data transfers and usage transactions providing assurance that only transactions conforming to the contract policies are authorized and registered in the ledger. The data usage control policies are based on "Event-Condition-Action" rule. An "Event" part is a pattern matching expression of actual or tentative actions, A "Condition" is a complex expression with propositional, temporal, and cardinality operators. An "Action" is an enforcement (allow, deny, modify, or delay) or reactive activity. Examples of these data usage policies are "whenever <My data is about to be accessed> and <I did not give my consent>, then "Deny".

Transactions are generated when a data usage contract is executed to obtain the desired data by data controllers or data processors. With transactions, data owners can trace the use of their data.

Figure 3 shows a data owner centric access and sharing type. The data owner creates a contract tailored for each controller that manages his/her data. The contract keeps track of the data shared with the data controller, the policies regulating the use of the data, and registers the data usage events representing the activities performed by the data controller using the data owner's data as input.

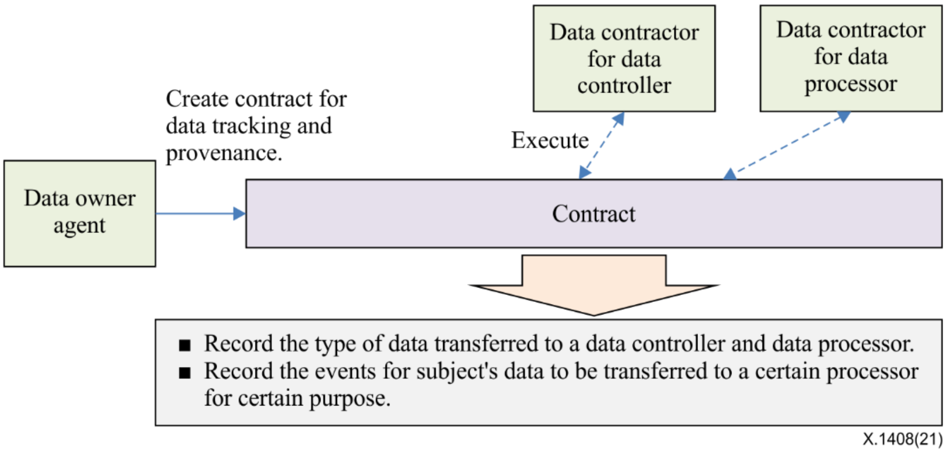
Data contractor for data controller
Data contractor for data processor
Create contract for data tracking and provenance.
Execute
Data owner agent
Contract
■ Record the type of data transferred to a data controller and data processor.
■ Record the events for subject's data to be transferred to a certain processor for certain purpose.
X.1408(21)

**Figure 3 – Data owner centric data access and sharing**

## 7.2 Transaction flow

### 7.2.1 Transaction flow of data processing without distributed ledger technology

This clause illustrates the sequence of the interactions between the entities in a current data processing model which is not using DLT.

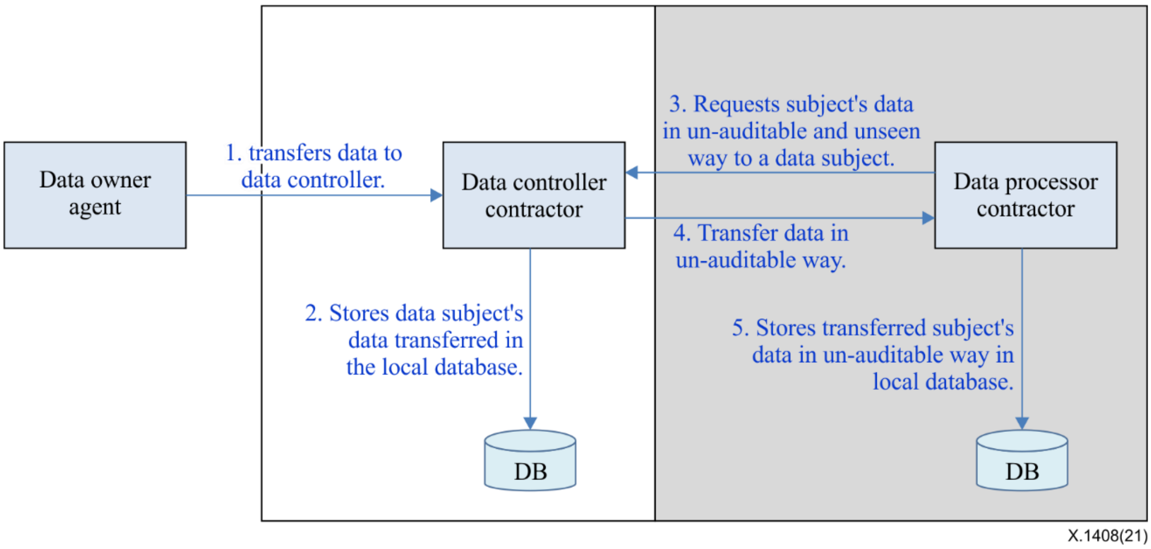
Data owner agent
1. transfers data to data controller.
Data controller contractor
3. Requests subject's data in un-auditable and unseen way to a data subject.
Data processor contractor
4. Transfer data in un-auditable way.
2. Stores data subject's data transferred in the local database.
5. Stores transferred subject's data in un-auditable way in local database.
DB
DB
X.1408(21)

**Figure 4 – Sequence of the interactions between the entities without distributed ledger technology**

It is assumed that the data owner is a person who receives a service, the data controller is a service provider that accesses or uses data of the data owner to provide a service and the data processor is another service provider that processes data on behalf of the data controller.

The data processing flow among data owner, data controller, and data processor in Figure 4 is as follows:

1) A data owner subscribes for service to a data controller. The data owner transfers its data to the data controller directly in a secure manner, e.g., through a secure tunnel with confidentiality and integrity protection, according to its data transfer policy.

2) The data controller receives the data and stores it in the local database according to its local data storage policy. The data controller will use this data according to a data usage policy set by the data controller.

3) The data processor requests a transfer of a specific data according to contract between the data controller and data processor. Its transfer of the data owner's data depends on the data transfer policy and consent of the data owner or provisions in the notice.

4) The data controller transfers a specific data owner's data to the data processor in a secure manner according to its data transfer policy.

5) The data processor stores the transferred data to its local database. A data processor will use the data owner's data for the interest of a data controller.

The configuration above has the following drawbacks:

- A data owner can request to change policy (for example, cancel transfer data to a certain processor), but he could not verify whether this change is implemented.
- A data owner has no means to know how its data is being processed, stored, and transferred.

### 7.2.2 Transaction flow of data processing with distributed ledger technology

This clause illustrates the sequence of the interactions between the entities in data access and sharing based on distributed ledgers. The data owner subscribes the data controller services, creates the data usage contract for this data controller regulating the use of his/her data, and transfers the data to the data controller.

For each new established contract, the data owner uses a new address to prevent linkability of the contracts established with each data controller and the need to maintain a list of all addresses used and the respective nonce established with each data controller or data processor. After creating a data usage contract, the data owner transfers the data to the data controller. The data usage contract refers to a policy-based smart contract that specifies constraints on the usage and redistribution of any data obtained explicitly or implicitly by the data controller.

Figure 5 shows the sequence of the interactions between the entities in the DLT based configuration.

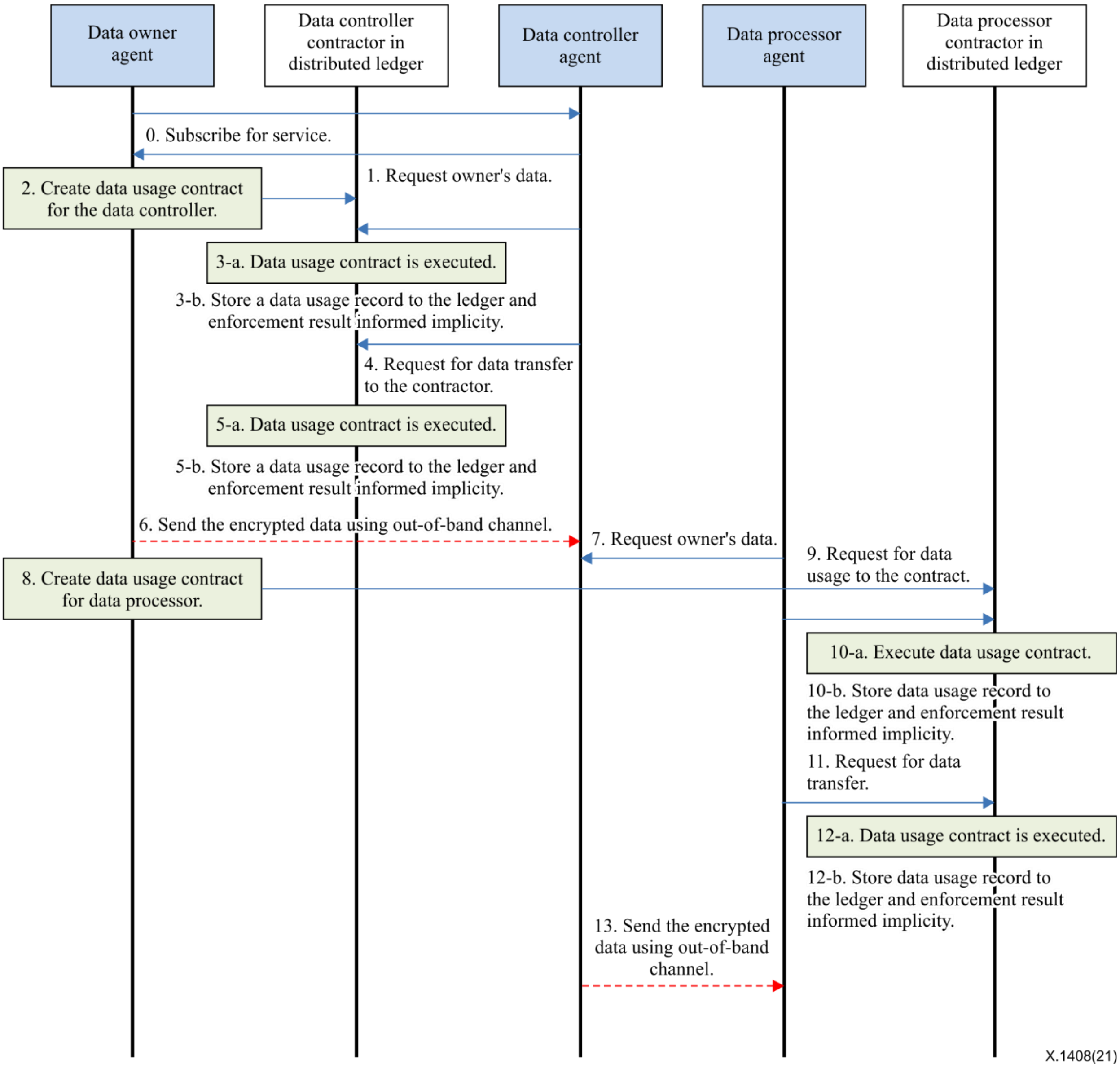
Data owner agent
Data controller contractor in distributed ledger
Data controller agent
Data processor agent
Data processor contractor in distributed ledger
0. Subscribe for service.
1. Request owner's data.
2. Create data usage contract for the data controller.
3-a. Data usage contract is executed.
3-b. Store a data usage record to the ledger and enforcement result informed implicity.
4. Request for data transfer to the contractor.
5-a. Data usage contract is executed.
5-b. Store a data usage record to the ledger and enforcement result informed implicity.
6. Send the encrypted data using out-of-band channel.
7. Request owner's data.
8. Create data usage contract for data processor.
9. Request for data usage to the contract.
10-a. Execute data usage contract.
10-b. Store data usage record to the ledger and enforcement result informed implicity.
11. Request for data transfer.
12-a. Data usage contract is executed.
12-b. Store data usage record to the ledger and enforcement result informed implicity.
13. Send the encrypted data using out-of-band channel.
X.1408(21)

**Figure 5 – Sequence of the interactions between the entities in the DLT based configuration**

The transaction flow of data access and sharing based on distributed ledger is as follows:

0) A data owner subscribes for service provided by a data controller after mutual identification and authentication.

1) The data controller agent requests the data owner's data.

2) The data owner agent creates a data usage contract for the data controller, known as a data controller contractor. The data usage contract should contain a list of data transferred to the data controller, such as data instantiation and instance value, and a policy that enables data

controller or data processor to check if their intended activity is allowed according to policy.

3-a) When the data controller desires to use data according to the contract, the data controller agent retrieves the data usage contract created by a data owner. Data controller agent executes the data usage contract to request access to the desired data.

3-b) The execution event is recorded to distributed ledger as a transaction, and if data controller agent is authorized, a data usage contract stores the enforcement event to the data owner agent according to the policy written in the data usage contract.

4) A data controller requests for data transfer to the data contractor.

5-a) A data controller agent executes data usage contract.

5-b) The data usage record is storage in the ledger.

6) The data owner agent sends the encrypted data of the data owner along with the enforcement event, through a separate channel for the data transmit in a secure manner where protection of confidentiality, integrity and data origin authentication is provided.

7) A data processor agent requests the data owner's data to the data controller agent after mutual identification and authentication.

8) The data owner agent creates a new data usage contract for the data processor. The data usage contract should contain a list of data transferred to the data processor, such as data instantiation and instance value, and a policy that enables data controller or data processor to check if their intended activity is allowed according to policy.

9) When data processor desires to use data according to the contract, data processor agent retrieves its data usage contract.

10-a) A data processor agent executes the data usage contract to request access to the desired data.

10-b) The execution event is recorded to distributed ledger as a transaction, and if data processor agent is authorized, data usage contract sends the enforcement event to the data controller agent according to the policy written in the data usage contract.

11) The data processor agent requests for data transfer from data controller.

12-a) The data processor agent executes data usage contract.

12-b) The data usage record is stored in the ledger.

13) The data controller agent sends the encrypted data of data owner along with the enforcement event, through separate channel for data transmit in a secure manner.

A technical specification based on the functional model to implement the data access and sharing is described in Annex B. See Annex B.2 for procedures of data access and sharing in implemented data access and sharing.

## 8 Security threats

This Recommendation refers to [b-ITU-T X.1401] for general threats to DLT and identifies threats specific to data access and sharing based on DLT in the following clauses.

### 8.1 Threat assumptions

It is assumed that the adversary has the following capabilities:

- to read, send and drop a transaction addressed to the distributed ledgers;
- to impersonate a data owner to give a data controller some rights without the proper data owner's consent;
- to link a data usage contract's identifier or a data usage contract to a specific data owner;

- to prevent the publication of a legitimate transaction. For example, in order to prevent data transfer notification to the data owner, an attacker may conduct a DoS attack against a data usage event or attempt a flooding attack on the distributed ledger with invalid data usage information.

## 8.2 Security threats to entities

### 8.2.1 Security threats to data controller

Threats to data controller in the data access and sharing based on DLT are identified as follows:

- Impersonation of data controller to data owner: Attacker can impersonate legitimate data controller.
- Impersonation of data controller to data processor: Attacker can impersonate legitimate data controller.
- Loss or theft of authentication key: Data controller can fail to authenticate as appropriate user by loss or theft of authentication key.
- Unintended use/access/action: Data usage contract can perform unintended action against data owner's or data controller's will because of faulty implementation of contracts.

### 8.2.2 Security threats to data processor

Threats to data processor in the data access and sharing based on DLT are identified as follows:

- Impersonation of data processor to data owner: Attacker can impersonate legitimate data processor.
- Impersonation of data processor to data controller: Attacker can impersonate legitimate data processor.
- Loss or theft of authentication key: Data processor can fail to authenticate as appropriate user by loss or theft of authentication key.
- Unintended use/access/action: Data usage contract can perform unintended action against data processor's will because of faulty implementation of contracts.

### 8.2.3 Security threats to data owner

Threats to data owner in the data access and sharing based on DLT are identified as follows:

Impersonation of data owner to data controller: Attacker impersonates legitimate data owner.

- The identity of the data owner could be identified.
- The owner cannot audit the history of processing by data controller or data processor of his or her data.
- Loss or theft of authentication key: Data owner can fail to authenticate as appropriate user by loss or theft of authentication key.
- Unintended use/access/action: Data usage contract can perform unintended action against data owner's will because of faulty implementation of contracts.

### 8.2.4 Security threats to data usage contract

Threats to data usage contract in the data access and sharing based on DLT are identified as follows:

- Leakage of PII data: Attacker can use vulnerabilities of the data usage contract to leak PII data from the data stored in that data usage contract.
- Identification of data usage contract generated by same data owner: Attacker can identify the data usage contract generated by same data owner.

- Unauthorized data transfer from data controller to data processor: Attacker can transfer the data from one data controller to another data processor who has no contract with data owner due to vulnerabilities of data usage contract.

### 8.3 Security threats to the communication between entities

Threats to the communications between entities are identified as follows:

- destruction of information;
- corruption or modification of information;
- theft, removal or loss of information;
- disclosure of information;
- interruption of services.

### 8.4 Security threats to data

#### 8.4.1 Security threats on inappropriate data processing related to data owner

Threats to the data stored in the distributed ledger are identified as follows:

- Disclosure of data owner's identity: Data stored in the distributed ledger could be used to identify the identity of data owner.
- Disclosure of transactions: Data stored in the distributed ledger could be used to disclose transactions between the data owner, data controller, data processors and data usage contracts.

#### 8.4.2 Security threats to data stored in on-chain or off-chain distributed ledgers

Threats to data on the ledger in the data access and sharing based on DLT are identified as follows:

- destruction of sensitive data on-chain or off-chain;
- corruption or modification of sensitive data on-chain or off-chain;
- theft, removal or loss of sensitive data on-chain or off-chain;
- disclosure of sensitive data on-chain or off-chain;
- identity of data owner is compromised using PII data on-chain or off-chain.

## 9 Security requirements for DLT based data access and sharing

This Recommendation refers to [b-ITU-T X.1402] for general security requirements for DLT and specifies security requirements specific to data access and sharing based on DLT in the following clauses.

### 9.1 General security requirements and recommendations for data access and sharing

The following general security requirements apply to data access and sharing based on DLT:

- It is required to do mutual authentication between the entities of the data access and sharing based on DLT before they start to communicate.
- It is required to provide integrity protection for data communication between the entities of data access and sharing based on DLT.
- It is required to provide confidentiality protection for data communication between the entities of data access and sharing based on DLT.
- It is required to prevent unauthorized disclosure of both PII data and shared data between data owners and data controller/processors.

- It is required to provide a key/certificate management for confidentiality and integrity protection for off-chain based data communications between the entities of data access and sharing based on DLT.
- It is recommended to allow auditing authorities to lead an investigation and obtain consistent proofs in case of non-compliant activities.
- It is recommended for each data owner to have a transparent view over how data are collected, accessed and processed.

The following security requirements and recommendations related to PII protection apply to data access and sharing:

- It is prohibited from containing PII as data in the on-chain distributed ledgers.
- PII data is required to be transferred using a separate secure communication channel.
- PII data about a data owner is required to be stored off-chain.
- Only non-PII data, such as only hashed data values and enciphered PII, are required to be on-chain.
- It is required for a data controller or data processor to be able to prove the data is stored and processed according to the data owners' data protection requirements, in case a data breach happens.
- It is required to provide/implement secure key management methods and support secure key backup.
- It is recommended for the data access and sharing to record data containing PII off-chain instead of main chain in cases where data contains PII.
- It is recommended for the data access and sharing to store in the main chain of the distributed ledgers the metadata as hashed value to maintain the distributed ledgers, which is referenced to data stored off-chain.
- It is recommended for the data access and sharing to transform a data owner's identity information into pseudonym using de-identification techniques [b-ISO/IEC 20889], which are stored on-chain.
- It is recommended to ban/blacklist misbehaving users on the DLT platform.

## 9.2 Security requirements to agents

### 9.2.1 Security requirements and recommendations for data owner agent

The following security requirements apply to the data owner agent:

- It is required for the data owner agent to maintain a list of all addresses used and the respective nonce established with each data controller or processor.
- It is required that data owner agent authenticates data controller.
- It is required that data owner agent supports the credential (e.g., certificate, pre-shared key) management.
- It is recommended for the data owner to create a contract tailored for each controller that manages his/her data.
- It is recommended to keep track of the data shared with the data controller and the policies regulating the use of the data, and register the data usage events representing the activities performed by the data controller using the data owner data as input.
- PII data about the data owner is required to be de-identified or encrypted when they are stored on-chain.

- PII data about the data owner is required to be stored off-chain with a strong authorization mechanism.
- It is recommended that the data owner agent supports data encryption.

### 9.2.2 Security requirements and recommendations for data controller agent

The following security requirements apply to the data controller agent:

- It is required for the data controller agent to be authenticated to the data owner.
- It is required for the data controller agent to authenticate the data owner.
- It is required for the data controller agent to authenticate the data processor.
- It is required that the data controller agent checks if this data usage activity that wants to perform a data usage activity is allowed according to the policies.
- It is required that the data controller agent authorizes the data owner agent to share her/his data.
- It is required that the data controller agent authorizes the data processor agent to share the data.
- It is required that the data controller agent supports the credential (e.g., certificate, pre-shared key) management.
- It is required for the data controller agent to be responsible for its key management.
- It is required that the data controller agent supports anti-DoS protection.
- It is required that the data controller agent supports log and audit.
- It is recommended that the data controller agent provides confidentiality protection for the shared data which is stored in the system based on DLT.
- It is recommended that the data controller agent supports data encryption.
- It is recommended that the data controller agent supports hardening the operating system.
- It is recommended that the data controller agent supports hardware management to discover hardware failure automatically and recover from such a failure as soon as possible.

### 9.2.3 Security requirements and recommendations for data processor agent

The following security requirements apply to the data processor agent:

- It is required that the data processor agent authenticates the data controller.
- The data processor is required to be authenticated.
- It is required for the data processor agent to be authenticated by the data controller.
- It is required that the data processor agent supports the credential (e.g., certificate, pre-shared key) management.
- It is required for the data processor agent to be responsible for its key management.
- It is recommended to audit logs of its contract to detect misbehaviour.
- It is recommended that the data processor agent supports data encryption.

## 9.3 Security requirements for communication between agents

### 9.3.1 Security requirements for communication between data owner agent and data controller agent

The following security requirements and recommendations apply to communication between the data owner agent and the data controller agent:

- It is required to prevent unauthorized disclosure of shared data between the data owner agent and the data controller agent.

- It is required to perform mutual authentication between the data owner agent and the data controller agent.
- It is required for the data controller agent to authorize the data owner agent to share her/his data.
- It is required to provide replay protection for the communication between the data owner agent and the data controller agent.
- It is recommended to provide data confidentiality protection for the communication between the data owner agent and the data controller agent.
- It is recommended to provide data integrity protection for the communication between the data owner agent and the data controller agent.

### 9.3.2 Security requirements and recommendations for communication between data controller agent and data processor agent

The following security requirements apply to communication between the data controller agent and the data processor agent:

- It is required to prevent unauthorized disclosure of shared data between the data controller agent and the data processor agent.
- It is required to perform mutual authentication between the data controller agent and the data processor agent.
- It is required for the data controller agent to authorize the data processor agent to consume the shared data.
- It is required to provide replay protection for the communication between the data controller agent and the data processor agent.
- It is recommended to provide data confidentiality protection for the communication between the data controller agent and the data processor agent.
- It is recommended to provide data integrity protection for the communication between the data controller agent and the data processor agent.

## 9.4 Security requirements for data

### 9.4.1 Security requirements and recommendations for data usage contract

The following security requirements apply to the data usage contract:

- It is required to use data usage contracts to encode data provenance information and data preference requirements that enable data subjects to evaluate who has accessed their data and to specify the conditions for storage, processing, and transferring of the data.
- It is required to use data usage contracts to store a list of the data transferred to the data controller agent including data instantiation and instance value, e.g., e-mail: 'user@host.com'.
- It is required to obfuscate the provenance information in the data usage contract by using the hash function and random number, known as a nonce, which is shared with the data controller agent.
- It is required for the data usage contract to have an authorization policy to transfer data from data controller to data processor, which is created by a data owner.
- It is required to obfuscate the activity and attributes using the hash function and the random number.
- It is required for the data usage contract to use a new DLT address for each new established contract, to prevent linkability of the contracts of each data owner which is established with each controller.

- It is required for the data usage contract to be developed to comply with secure design and secure coding practices as defined in [b-ISO/IEC 27034-3].
- It is required for the data usage contract to be secure against known vulnerabilities, unsafe functions, and unsafe libraries.
- It is required for the data usage contract to be checked with open source based publicly available security audit tool. Code review using manual and/or automated tools is recommended to be conducted.
- It is required to perform secure authentication when permission is requested.
- It is required to grant access to the appropriate authenticator according to the contract.
- It is required to expire a contract that grants access appropriately in a timely manner according to the contract.
- It is required to enable termination of a contract at any time that the data owner desires.
- It is recommended to preserve the integrity of the data usage contract.
- It is required for the data usage contractor to authenticate the data controller agent and data processor agent that has a contract with the data owner agent.

### 9.4.2 Security requirements and recommendations to protect data on-chain or off-chain

The following security requirements apply to protect data on-chain or off-chain:

- It is required that only an authorized user accesses the data on-chain or off-chain.
- It is required to provide continuous availability for the data on-chain or off-chain.
- It is recommended to provide confidentiality for data on-chain or off-chain.

# Annex A

## Other types of data access and sharing models

(This annex forms an integral part of this Recommendation.)

This annex describes other types of data access and sharing models. Figure A.1 shows a conceptual data centric access and sharing model for each data owner. The data owner creates a generic contract for each data instance that is shared for all data controllers accessing the data. The contract contains the list of data controllers that were given access to a particular data instance.

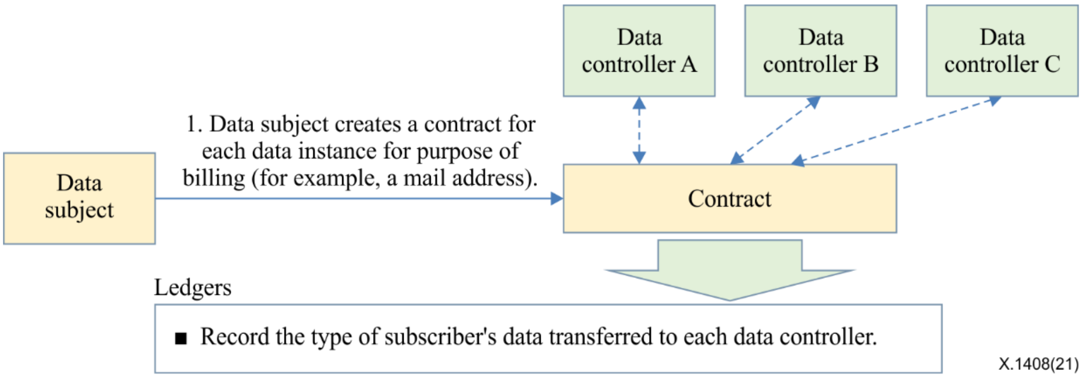
Data controller A
Data controller B
Data controller C
1. Data subject creates a contract for each data instance for purpose of billing (for example, a mail address).
Data subject
Contract
Ledgers
■ Record the type of subscriber's data transferred to each data controller.
X.1408(21)

**Figure A.1 – Data centric access and sharing model for data owner**

Figure A.2 shows this conceptual data controller centric access and sharing model. The data controller creates a contract for multiple data owners, which specifies how the data received from all data owners is treated. A data owner then joins the contract in case they accept the data usage policies of the controller, similar to the Platform for Privacy Preferences Project (P3P) [b-P3P].

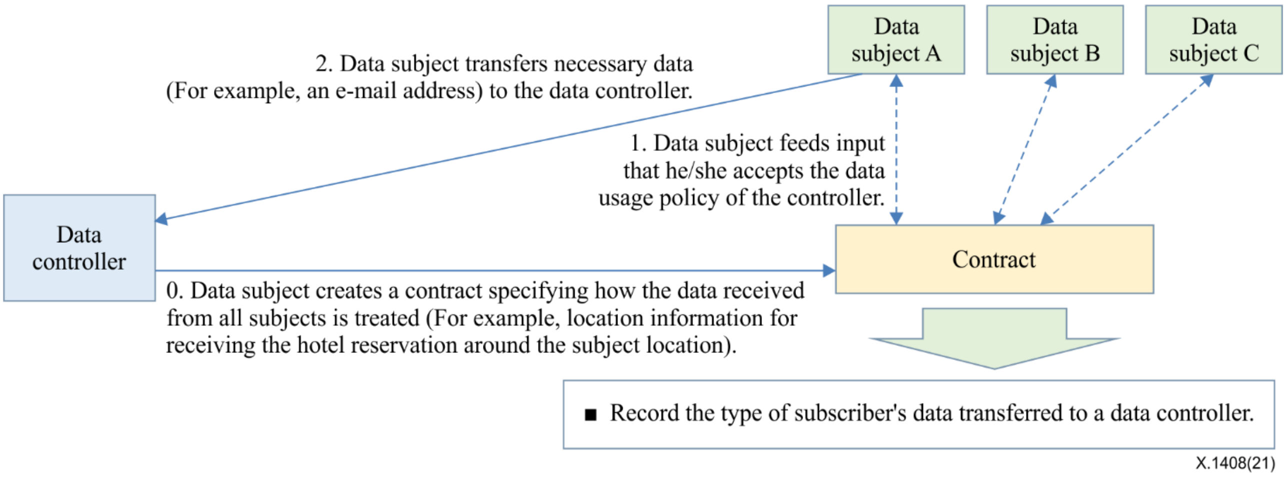
Data subject A
Data subject B
Data subject C
2. Data subject transfers necessary data (For example, an e-mail address) to the data controller.
1. Data subject feeds input that he/she accepts the data usage policy of the controller.
Data controller
Contract
0. Data subject creates a contract specifying how the data received from all subjects is treated (For example, location information for receiving the hotel reservation around the subject location).
■ Record the type of subscriber's data transferred to a data controller.
X.1408(21)

**Figure A.2 – Data controller centric access and sharing model**

# Annex B

## Technical specification to implement the functional model

(This annex forms an integral part of this Recommendation.)

### B.1 Components of functional architecture

According to the general model defined in clause 7, Figure B.1 shows the components of the functional architecture of data access and sharing model based on the distributed ledger technology.

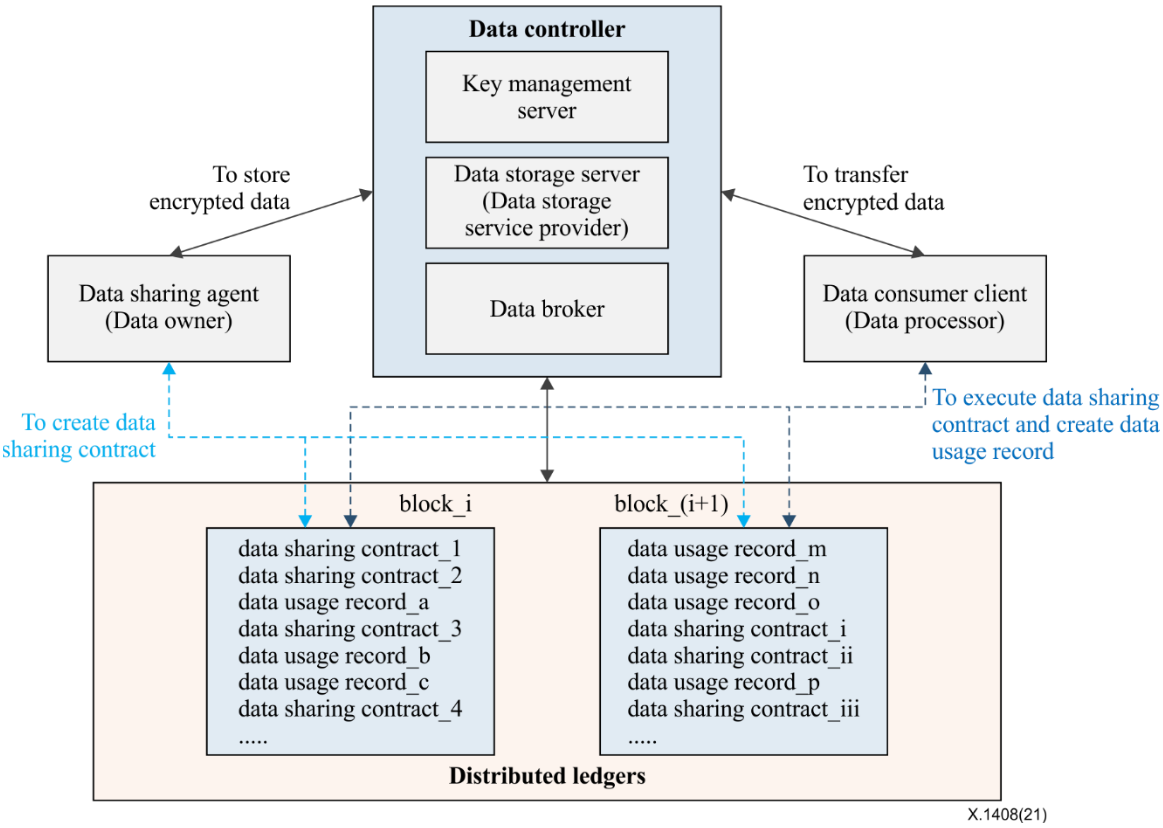

**Figure B.1 – Functional architecture for data access and sharing**

The components of functional architecture in Figure B.1 are described below (*italics* are used to indicate logical functions in Figure B.1):

- *Data sharing agent* representing the component data owner agent in Figure 1 of clause 7.1, is an application to collect raw data according to data collection format, to make the data anonymous, to encrypt the collected data and then store it as cipher text at a data storage server, to define data access policy which specifies constraints on the usage and redistribution of any data obtained explicitly or implicitly, to create a policy-based data usage contract request and send the request to a data broker.
- *Data consumer client* representing data processor *(e.g., service providers or other third parties who do further data analysis or use data to promote their products)* is an application and requests to execute data usage contracts published by a data broker in order to obtain the desired data. The data broker executes the data usage contract and creates transactions recording data usage, then authorizes a data consumer client to access the data stored at a data storage server.

- The components *Data broker* and *Key management server* together with *Data storage server* in Figure B.1 represent the component data controller agent in Figure 1 of clause 7.1. The *Data storage server* and *key management server* are introduced to provide confidentiality protection and integrity protection for the data to be shared.
  - *Data broker* is an application, which is implemented based on a common distributed ledger platform to enable data owners to share their data with data consumers (i.e., data processors) securely and automatically. *Data broker,* acting as data transaction coordinator between data owners and data consumers, has the following capabilities: 1) to help to design the data collection format based on a data consumer's requirements and some potential application scenarios; 2) to manage the registration of data owners and data consumers; 3) to create distributed-ledger-based data usage contracts which includes data access policy defined by data owners; 4) to execute data usage contracts according to the data access requests from data consumers; 5) to generate and keep transactions to enable data owners to track the usage of their data by data consumers; 6) to issue authorization tokens which authorize data consumers to access the data of data owners.
  - *Data storage server* represents data storage service providers to store the data for data owners. The data is encrypted and stored as ciphertext at the *Data storage server* and will not be exposed to data storage service providers, since the encryption key is not stored together with the data in the same server. The *Data storage server* has the following capabilities: 1) to receive and store the data from the *Data sharing agent*; 2) to receive the request for accessing data from the *Data consumer client*; 3) to send the requested data to the *Data consumer client*.
  - *Key management server* is introduced in order to manage the encryption key and has the following capabilities: 1) to receive and store the encryption key for the data to be shared from the *Data sharing agent*; 2) to receive the authorization token from the *Data consumer client*; 3) to validate the received authorization token with the help of the *Data broker*; 4) to send the encryption key to the *Data consumer client*.

## B.2 Procedures of data access and sharing based on DLT

Figure B.2 shows the procedures of users to manage their data access and sharing based on distributed ledgers. There are two stages for users to collect, store and share their data. The first stage is to prepare for data sharing which shows as black texts with black lines (i.e., steps 1-3) in Figure B.2. The second stage is to do data sharing or data consuming which shows as blue texts with blue lines (i.e., steps 4-10) in Figure B.2.

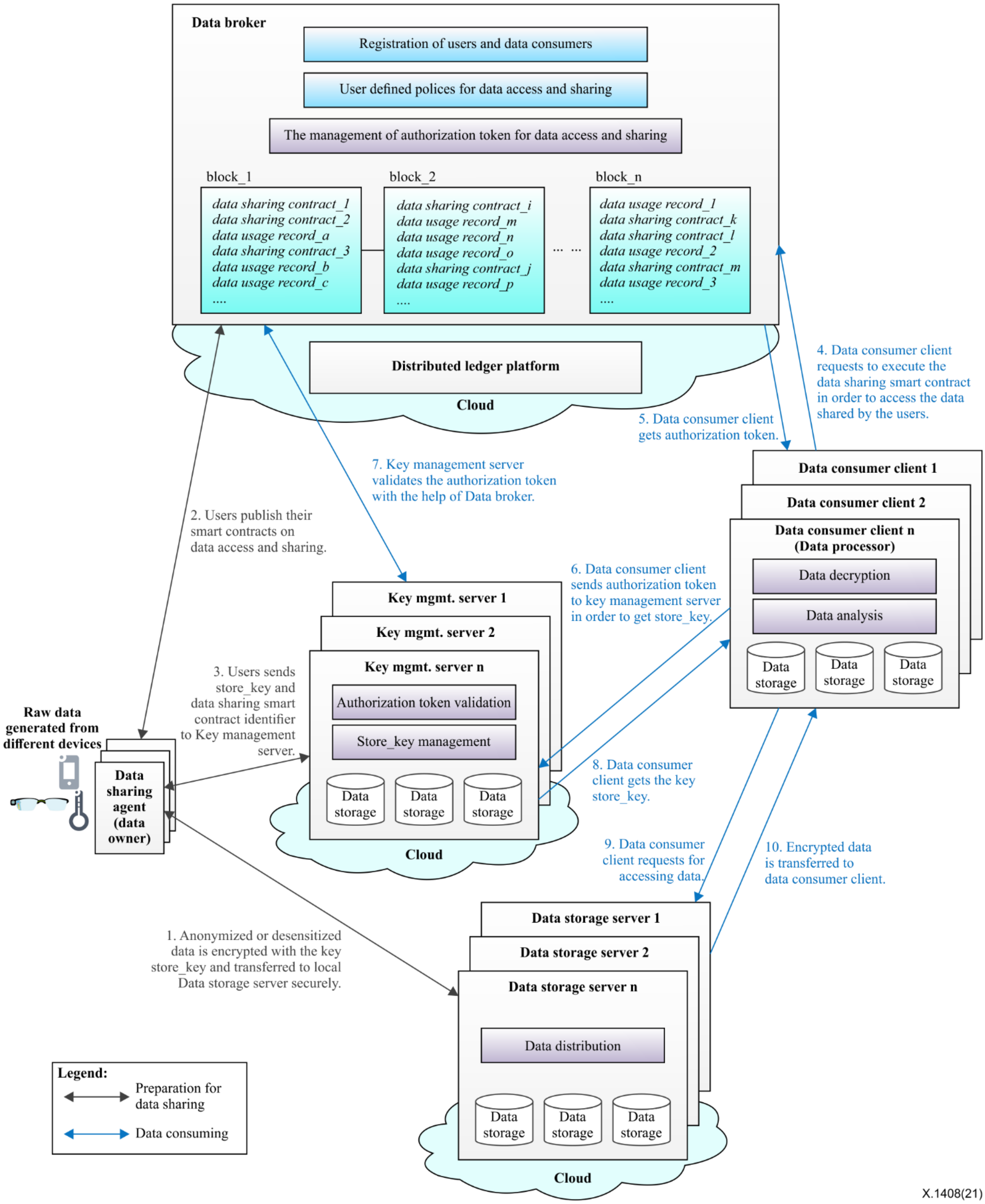
Data broker
Registration of users and data consumers
User defined polices for data access and sharing
The management of authorization token for data access and sharing
block_1
data sharing contract_1
data sharing contract_2
data usage record_a
data sharing contract_3
data usage record_b
data usage record_c
....
block_2
data sharing contract_i
data usage record_m
data usage record_n
data usage record_o
data sharing contract_j
data usage record_p
....
block_n
data usage record_1
data sharing contract_k
data sharing contract_l
data usage record_2
data sharing contract_m
data usage record_3
....
Distributed ledger platform
Cloud
4. Data consumer client requests to execute the data sharing smart contract in order to access the data shared by the users.
5. Data consumer client gets authorization token.
7. Key management server validates the authorization token with the help of Data broker.
2. Users publish their smart contracts on data access and sharing.
Data consumer client 1
Data consumer client 2
Data consumer client n (Data processor)
Data decryption
Data analysis
Data storage
Data storage
Data storage
6. Data consumer client sends authorization token to key management server in order to get store_key.
Key mgmt. server 1
Key mgmt. server 2
Key mgmt. server n
Authorization token validation
Store_key management
Data storage
Data storage
Data storage
Cloud
3. Users sends store_key and data sharing smart contract identifier to Key management server.
Raw data generated from different devices
Data sharing agent (data owner)
8. Data consumer client gets the key store_key.
9. Data consumer client requests for accessing data.
10. Encrypted data is transferred to data consumer client.
Data storage server 1
Data storage server 2
Data storage server n
Data distribution
Data storage
Data storage
Data storage
Cloud
1. Anonymized or desensitized data is encrypted with the key store_key and transferred to local Data storage server securely.
Legend:
Preparation for data sharing
Data consuming
X.1408(21)

**Figure B.2 – Procedures of data access and sharing based on DLT**

The procedures of users (i.e., data owners) to manage their data access and sharing based on distributed ledgers shown in Figure B.2 are described as below (*italics* are used to indicate logical functions in Figure B.2):

Stage one: To prepare for data sharing, which is shown as black texts with black lines (i.e., steps 1-3) in Figure B.2.

1) According to the data collection form (which is designed by the *Data broker*), the *Data sharing agent* collects raw data of users, anonymizes and desensitizes the collected raw data, then encrypts the anonymized and desensitized data with the key store_key and sends the encrypted data to the *Data storage server* securely. The integrity verification code of the data should be stored together with the encrypted data. The integrity verification code is generated by signing the hash value of the data with the user's private key. The mechanisms to secure the transportation of data between the *Data sharing agent* and the *Data storage server* can refer to existing security mechanisms such as transport level security (TLS) and Internet protocol security (IPsec).

2) The user (i.e., data owner) registers herself/himself in the data broker with her/his public key. The *Data sharing agent* (representing the user) sends the user's data sharing request to the *Data broker*. The data sharing request includes user's identifier, user's public key, data description, data usage policy (e.g., to indicate that the access and sharing of the data should meet some specific requirements, such as the location of the data consumer (i.e., data processor), the industry segment (e.g., finance, healthcare/pharma, smart city, energy/utilities) of the data consumer, the country of the data consumer, etc.), and the address of user's data stored at the *Data storage server*. After receiving such a request, the *Data broker* creates a data usage contract for the user to regulate the use of his/her data (e.g., how to share the data with a data consumer). Each data usage contract describes how to access and share the data according to the data usage policy defined by the user. The data usage contract has its own unique identifier, which is derived from the user's public key, the address of user's data, and optionally with other parameters (e.g., time stamp, data broker identifier) with using a cryptographic hash function. *The Data broker* publishes the data usage contract and sends the smart contract identifier to the *Data sharing agent*.

3) The *Data sharing agent* sends the data usage contract identifier and the key store_key to the *Key management server*. The *Key management server* keeps the smart contract identifier together with the corresponding key store_key securely. The mechanisms to secure the transportation of the key store_key and data sharing smart contract identifier between the *Data sharing agent* and the *Key management server* can refer to existing security mechanisms such as TLS and IPsec.

   In order to improve the security of the collected data, the collected data is encrypted and stored separately with the encryption key. The encrypted data is stored at the *Data storage server* and the encryption key is stored at the *Key management server*. The *Key management server* does not know the collected data even though it knows the key store_key, because the key *management server* does not know the address of user's data. The *Data storage server* does not know user's data either since the data is encrypted with the key store_key. The *Data storage server* does not know the key store_key.

Stage two: To do data sharing or data consuming, which is shown as blue texts with blue lines (i.e., steps 4-10) in Figure B.2.

4) Data consumer registers herself/himself in the *Data broker*. Data consumer finds the desired data from the *Data broker*. The *Data consumer client* (representing data consumer) sends the request to the *Data broker* to execute the corresponding data usage contract. This request includes the data consumer's identifier (e.g., public key) and data usage contract identifier.

5) The *Data broker* executes the data usage contract and issues an authorization token to the *Data consumer client*. The authorization token includes the data consumer identifier, data usage contract identifier, and the address of the *Key management server*. The *Data broker* creates a transaction to keep/record the data access/usage, data consumer identifier, and data usage contract identifier. Each transaction has its own unique identifier, which is derived from the smart contract identifier, data consumer identifier, authorization token, data usage, and optionally with other parameters (e.g., time stamp, data broker identifier) with using a cryptographic hash function. Based on the transactions, users (i.e., data owners) can track the usage of their data. The authorization token and the address of user's data optionally together with user's public are sent to the *Data consumer client*.

6) The *Data consumer client* sends authorization token to the *Key management server*. The mechanisms to secure the transportation of authorization token between the *Data consumer client* and the *Key management server* can refer to existing security mechanisms such as TLS and IPsec.

7) The *Key management server* validates the received authorization token with the help of the *Data broker*.

8) Based on the data usage contract identifier, the *Key management server* looks for the corresponding key store_key and sends it to the *Data consumer client*. The mechanisms to secure the transportation of the key store_key between the *Data consumer client* and the *Key management server* can refer to existing security mechanisms such as TLS and IPsec.

9) The *Data consumer client* sends the request to the *Data storage server* to fetch the encrypted data.

10) The *Data storage server* sends the encrypted data together with data integrity verification code to the *Data consumer client*. The *Data consumer client* will get the plain-text data with the key store_key gotten from the *Key management server*. The *Data consumer client* can verify the data integrity with the data integrity verification code and the user's public key.

One block/ledger stored at the *Data broker* in Figure B.2 can store one or more data usage contracts and/or one or more transactions.

Based on the data usage records, the data owners can easily check all data transfers and usage transactions providing assurance that only transactions conforming to the contract policies are authorized and recorded in the distributed ledgers.

# Bibliography

[b-ITU-T X.800] Recommendation ITU-T X.800 (1991), *Security architecture for Open Systems Interconnection for CCITT applications*.

[b-ITU-T X.1400] Recommendation ITU-T X.1400 (2020), *Terms and definitions for distributed ledger technology*.

[b-ITU-T X.1401] Recommendation ITU-T X.1401 (2019), *Security threats of distributed ledger technology*.

[b-ITU-T X.1402] Recommendation ITU-T X.1402 (2020), *Security framework for distributed ledger technology*.

[b-DFS-Glossary] ITU-T Focus Group Digital Financial Services, *Digital Financial Services (DFS) Glossary*, 2017.1.

[b-ENISA] ENISA (2016), *Distributed Ledger Technology & Cybersecurity Improving information security in the financial sector*, December.

[b-FG DLT I-038] DLT-I-038, *European Financial Transparency Gateway / Energy distribution with the use of smart contracts / Smart contracts for data accountability and provenance tracking*.

[b-FG DLT D4.1] DLT D4.1, *Distributed ledger technology regulatory framework*.

[b-ISO/IEC 20889] ISO/IEC 20889:2018, *Privacy enhancing data de-identification terminology and classification of techniques*.

[b-ISO/IEC 27000] ISO/IEC 27000:2018, *Information technology – Security techniques – Information security management systems – Overview and vocabulary*.

[b-ISO/IEC 27034-3] ISO/IEC 27034-3:2018, *Information technology – Application security – Part 3: Application security management process*.

[b-ISO/IEC 29100] ISO/IEC 29100:2011, *Information technology – Security techniques – Privacy framework*.

[b-ISO/TR 14292] ISO/TR 14292:2012, *Health informatics – Personal health records – Definition, scope and context*.

[b-ISO/TR 23244] ISO/TR 23244, *Privacy and Personally Identifiable Information protection considerations*.

[b-Kaaniche] Nesrine Kaaniche, Maryline Laurent (2017), *A blockchain-based data usage auditing architecture with enhanced privacy and availability*, 16th IEEE International Symposium on Network Computing and Applications, pp.1-5, at https://www.computer.org/csdl/proceedings/nca/2017/1465/00/08171384.pdf.

[b-Neisse] Ricardo Neisse, Gary Steri, and Igor Nai-Fovino (2017), *A Blockchain-based Approach for Data Accountability and Provenance Tracking*, https://arxiv.org/pdf/1706.04507.pdf.

[b-P3P] Platform for Privacy Preferences (P3P) Project at https://www.w3.org/P3P/.

## SERIES OF ITU-T RECOMMENDATIONS

| | |
|---|---|
| Series A | Organization of the work of ITU-T |
| Series D | Tariff and accounting principles and international telecommunication/ICT economic and policy issues |
| Series E | Overall network operation, telephone service, service operation and human factors |
| Series F | Non-telephone telecommunication services |
| Series G | Transmission systems and media, digital systems and networks |
| Series H | Audiovisual and multimedia systems |
| Series I | Integrated services digital network |
| Series J | Cable networks and transmission of television, sound programme and other multimedia signals |
| Series K | Protection against interference |
| Series L | Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant |
| Series M | Telecommunication management, including TMN and network maintenance |
| Series N | Maintenance: international sound programme and television transmission circuits |
| Series O | Specifications of measuring equipment |
| Series P | Telephone transmission quality, telephone installations, local line networks |
| Series Q | Switching and signalling, and associated measurements and tests |
| Series R | Telegraph transmission |
| Series S | Telegraph services terminal equipment |
| Series T | Terminals for telematic services |
| Series U | Telegraph switching |
| Series V | Data communication over the telephone network |
| **Series X** | **Data networks, open system communications and security** |
| Series Y | Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities |
| Series Z | Languages and general software aspects for telecommunication systems |

Printed in Switzerland
Geneva, 2021