INTERNATIONAL STANDARD

ISO 23257

First edition
2022-02

Blockchain and distributed ledger technologies — Reference architecture

Technologies des chaînes de blocs et technologies de registre distribué — Architecture de référence

Reference number
ISO 23257:2022(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) is a worldwide federation of national standards bodies (ISO member bodies). The work of preparing International Standards is normally carried out through ISO technical committees. Each member body interested in a subject for which a technical committee has been established has the right to be represented on that committee. International organizations, governmental and non-governmental, in liaison with ISO, also take part in the work. ISO collaborates closely with the International Electrotechnical Commission (IEC) on all matters of electrotechnical standardization.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of ISO documents should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT), see www.iso.org/iso/foreword.html.

This document was prepared by Technical Committee ISO/TC 307, *Blockchain and distributed ledger technologies*.

Any feedback or questions on this document should be directed to the user’s national standards body. A complete listing of these bodies can be found at www.iso.org/members.html.

Introduction

Records of transactions, based on certain agreed upon conditions, form the basis for exchanging assets between parties. Businesses and governments have been operating for centuries using this foundation. While physical ledgers were once used, they have largely been replaced with modern technology. However, in traditional approaches, a ledger must be centrally controlled by one or a small number of parties, and other stakeholders must rely on them as agents to change those ledgers.

An important property of a ledger is verifiability. This means that the parties can verify that the set of transactions in the ledger is complete and accurate. As a result, these parties can identify irregularities in transactions, for example, to verify that digital assets of the participants are correctly accounted within a financial ledger. Currently, it is possible to achieve a verifiable ledger in a centralized way by making certain trust assumptions. However, verifiability can be also achieved by distributing the storage and decentralizing the control of the ledger with minimal trust in any one party.

By maintaining a ledger in a distributed network, Distributed Ledger Technology (DLT) systems, including blockchain systems, allow a much wider range of parties to have a shared view of the ledger and to make their own changes to that ledger.

A broad spectrum of DLT based business solutions is possible. This document presents a reference architecture for such DLT based solutions. It starts with the definitions and concepts of blockchain and DLT such as the system organization, nature of access, type of consensus and the roles and responsibilities of the participants. Given that the reference architecture must accommodate a wide variety of possible use cases, it touches upon various business domains and their respective use cases at a high level. Historically, ledgers have facilitated the exchange of assets, but DLT solutions can also be used more broadly for reporting, auditing, and coordination. The document finally presents the reader with various layers of a reference architecture for DLT systems and the functional components in the layers.

This document is relevant to, among other, academics, architects, customers, users, developers, regulators, auditors, and standards development organizations.

Blockchain and distributed ledger technologies — Reference architecture

1 Scope

This document specifies a reference architecture for Distributed Ledger Technology (DLT) systems including blockchain systems. The reference architecture addresses concepts, cross-cutting aspects, architectural considerations, and architecture views, including functional components, roles, activities, and their relationships for blockchain and DLT.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO 22739, *Blockchain and distributed ledger technologies — Vocabulary*

ISO/IEC 24760-1, *IT Security and Privacy — A framework for identity management — Part 1: Terminology and concepts*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 22739, ISO/IEC 24760-1 and the following apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

3.1
activity
specified pursuit or set of tasks

[SOURCE: ISO/IEC 17789:2014, 3.2.1]

3.2
architecture
fundamental concepts or properties of a system in its environment embodied in its elements, relationships, and in the principles of its design and evolution

[SOURCE: ISO/IEC/IEEE 42010:2011, 3.2]

3.3
behavioural interoperability
interoperability so that the actual result of the exchange achieves the expected outcome

[SOURCE: ISO/IEC 19941:2017, 3.1.6]

3.4
data archiving
digital preservation process that is moving data into a managed form of storage for long-term retention

[SOURCE: ISO 5127:2017, 3.1.11.19]

3.5
data flow
sequence in which data transfer, use, and transformation are performed during the execution of a computer program

[SOURCE: ISO/IEC/IEEE 24765:2017, 3.1006]

3.6
disruption
incident, whether anticipated (e.g. hurricane) or unanticipated (e.g. power failure/outage, earthquake, or attack on information and communication technology systems/infrastructure) which disrupts the normal course of operations at an organization's location

[SOURCE: ISO/IEC 27031:2011, 3.6]

3.7
distributed ledger technology governance
DLT governance
system for directing and controlling a distributed ledger technology system including the distribution of on-ledger and off-ledger decision rights, incentives, responsibilities and accountabilities

3.8
finality
property of a ledger that guarantees transactions in confirmed ledger records are irreversible and cannot be altered or deleted

3.9
functional component
functional building block needed to engage in an activity, backed by an implementation

[SOURCE: ISO/IEC 17789:2014, 3.2.3]

3.10
fungible
capable of mutual substitution among individual units

Note 1 to entry: The individual units can be digital assets, e.g. tokens.

3.11
governance
system of directing and controlling

[SOURCE: ISO/IEC 38500:2015, 2.8]

3.12
incident
anomalous or unexpected event, set of events, condition, or situation at any time during the life cycle of a project, product, service, or system

[SOURCE: ISO/IEC/IEEE 24748-1:2018, 3.22]

3.13
interoperability
ability of two or more systems or applications to exchange information and to mutually use the information that has been exchanged

[SOURCE: ISO/IEC 17788:2014, 3.1.5]

3.14
party
natural person or legal person, whether or not incorporated, or a group of either

[SOURCE: ISO/IEC 17789:2014, 7.2.3]

3.15
personally identifiable information
PII
information that (a) can be used to establish a link between the information and the natural person to whom such information relates, or (b) is or can be directly or indirectly linked to a natural person

Note 1 to entry: The “natural person” in the definition is the PII principal. To determine whether a PII principal is identifiable, account should be taken of all the means which can reasonably be used by the privacy stakeholder holding the data, or by any other party, to establish the link between the set of PII and the natural person.

[SOURCE: ISO/IEC 29100:2011/Amd 1:2018, 2.9]

3.16
policy interoperability
interoperability while complying with the legal, organizational and policy frameworks applicable to the participating systems

[SOURCE: ISO/IEC 19941:2017, 3.1.7]

3.17
provenance
information that documents the origin or source of an asset, any changes that have taken place since it was originated, and who has had custody of it since it was originated

[SOURCE: ISO/IEC 27050-1:2019, 3.19, modified — “Electronically Stored Information” has been replaced with “asset”.]

3.18
resilience
capability of a system to maintain its functions and structure in the face of internal and external change, and to degrade gracefully when this is necessary

[SOURCE: ISO 37101:2016, 3.33, modified — Definition replaced with text in Note 3 to entry, Notes to entry deleted.]

3.19
role
set of activities that serve a common purpose

[SOURCE: ISO/IEC 17789:2014, 3.2.7]

3.20
semantic data interoperability
interoperability so that the meaning of the data model within the context of a subject area is understood by the participating systems

[SOURCE: ISO/IEC 19941:2017, 3.1.5]

3.21
sub-role
subset of the activities of a given role

[SOURCE: ISO/IEC 17789:2014, 3.2.9]

3.22
syntactic interoperability
interoperability such that the formats of the exchanged information can be understood by the participating systems

[SOURCE: ISO/IEC 19941:2017, 3.1.4]

3.23
transport interoperability
interoperability where information exchange uses an established communication infrastructure between the participating systems

[SOURCE: ISO/IEC 19941:2017, 3.1.3]

3.24
smart contract
computer program stored in a DLT system wherein the outcome of any execution of the program is recorded on the distributed ledger

Note 1 to entry: A smart contract can represent terms in a contract in law and create a legally enforceable obligation under the legislation of an applicable jurisdiction.

[SOURCE: ISO 22739:2020, 3.72]

3.25
consensus
agreement among DLT nodes that 1) a transaction is validated and 2) the distributed ledger contains a consistent set and ordering of validated transactions

Note 1 to entry: Consensus does not necessarily mean that all DLT nodes agree.

Note 2 to entry: The details regarding consensus differ among DLT designs and this is a distinguishing characteristic between one design and another.

[SOURCE: ISO 22739:2020, 3.11]

4 Symbols and abbreviated terms

AMQP	Advanced Message Queuing Protocol
API	Application Programming Interface
CAdES	Cryptographic Message Syntax (CMS) Advanced Electronic Signature
DLT	Distributed Ledger Technology
DNS	Domain Name System
EDI	Electronic Data Interchange
FBA	Federated Byzantine Agreement
GDPR	General Data Protection Regulation
HTTP	Hyper Text Transfer Protocol

HTTPS	Hypertext Transfer Protocol Secure over Socket Layer
ICT	Information and Communication Technology
IDE	Interactive Development Environment
IoT	Internet of things
IPFS	InterPlanetary File System
JSON	JavaScript Object Notation
MQTT	Message Queuing Telemetry Transport
P2P	Peer-to-peer
PBFT	Practical Byzantine Fault Tolerance
PKI	Public Key Infrastructure
RA	Reference Architecture
XML	eXtensible Markup Language

5 Concepts

5.1 DLT and blockchain systems

5.1.1 General

To understand blockchain and DLT systems, it is necessary to provide a description of the essential concepts associated with these systems and the range of distributed ledger technologies that exist.

A ledger is a long-established concept used in business and technology. When applied to ICT systems, it is an information store that keeps “final and definitive” records of transactions. Ledgers were originally and principally applied to financial transactions and to accounting practices. However, ledgers can be used to record transactions of almost any type: for example, the movements and transfers of physical objects.

A highly desirable property of a ledger is tamper-resistance, i.e. that transaction records, once entered into the ledger, are difficult to alter by design, and they cannot be altered without the alteration being clearly evident on inspection, whether the alteration is deliberate or accidental, malicious or benign.

The word "tamper-resistant" is more appropriate than "tamper-proof" since it can be extremely hard to prevent all forms of tampering. Similarly, although immutability is a design goal of DLT systems, it cannot be absolutely guaranteed. The word "tamper-evident" can be applied to a system that has the desirable characteristic of enabling any unauthorized changes to be clearly visible.

A distributed ledger has its entries stored across a series of nodes in a network, rather than in a single location. For example, the different nodes in the network can be owned, and interacted with, by different parties where each party has its own instance of records of only the transactions that it is involved in. DLT systems are designed to implement distributed ledgers, which is a significant challenge due to the need to agree on and maintain the transaction records in the distributed ledger.

With DLT, consensus ensures that every replicated version of a transaction is the same across all the nodes where it is stored – and that its contents are generally agreed amongst the parties involved in the transaction. The set of records in the distributed ledger should be verifiable and auditable. One of the major goals of DLT is to provide non-repudiable online transaction records.

Looking at DLT in this way does not imply that every node in the network stores exactly the same set of transaction records (although that can be the case for some forms of distributed ledger). It also does not imply that every party that participates in the distributed ledger has access to all the transaction records (it is possible that parties do not have access to transaction records they are not involved in). The transaction records stored on a node in a DLT system can be a whole or partial set of the distributed ledger implemented by the DLT system.

5.1.2 Blockchain DLT and non-blockchain DLT

Blockchain systems are a subset of distributed ledger technologies in which the state of a distributed ledger is maintained by processing batches of transactions in cryptographically secured data structures known as blocks. A valid protocol should ensure that each block is cryptographically linked to an immediately previous block forming a unique sequence of blocks in time. The complete sequence of cryptographically associated blocks forms a globally accessible append-only data structure - the blockchain - that provides the canonical version of the global transaction history.

In order to ensure that the ledger update process results in a single ledger state for a given block, blockchain protocols include a consensus mechanism that provides a total ordering of all transactions within the block. The collective action of all nodes in the blockchain system functions as a timestamp server that validates pending transactions and updates the current ledger state by sequentially appending blocks to the blockchain.

To implement a distributed ledger, blockchain systems require a mechanism to distribute new blocks to all nodes, a mechanism to validate transactions, and a mechanism to ensure consistency of all the copies of the blockchain.

The word “blockchain” is commonly applied both to the data structure, and to the complete implementation of a distributed ledger that uses the blockchain data structure.

A blockchain data structure can be used to implement something that is not a distributed ledger; however, such uses of blockchain are not within the scope of this document. This includes a centralized implementation of a blockchain database.

Not all DLT systems are blockchain-based. Some DLT systems use different ledger data structures with different approaches to storing transaction records and maintaining integrity, which typically offer different characteristics (e.g. capability to support high transaction rates).

In some non-blockchain DLT systems, a transaction record is stored as a separate ledger entry rather than as a part of the content of a block. Also, the ledger structure can possibly not be a chain. For example, there are ledgers wherein the underlying structure is organized as a directed acyclic graph (DAG) with transactions represented by vertices. The use of a DAG for transactions can improve the time and cost required for transaction validation but increase the effort for synchronization.

5.2 Networking and Communications

Networking and communications are an essential part of DLT systems.

A distributed program, or distributed application is an application that runs on a distributed system.

A P2P distributed application architecture partitions tasks or workloads between peers. Peers are equally privileged, equally capable participants in the application. They form a P2P network of nodes.

A DLT network is a network of DLT nodes that make up a distributed ledger system. Usually, DLT nodes communicate via P2P networks.

The protocol chosen to communicate between DLT nodes depends on implementation options and considerations available.

5.3 DLT platform

A DLT platform is a set of processing, storage and communication entities which together provide the capabilities of the distributed ledger system on each DLT node.

A node in this case is a machine in a P2P network that runs software components that communicate to support the distributed ledger and which can store a replica of the ledger. The node can either be a physical machine or else some form of virtual execution environment such as one or more virtual machines or containers. Virtual environments can be used if the node is implemented using cloud computing, for example. See ISO/IEC TS 23167 for more information about virtual environments and containers.

A node storing a complete replica of the ledger is referred to as a full node. Some DLT systems have nodes which, by design, do not contain a complete replica of the ledger.

A blockchain platform is a DLT platform where the implementation technology is a blockchain.

The capabilities supported by a DLT platform can include

- secure runtime environments,
- smart contracts,
- ledger,
- transaction system,
- membership services,
- state management,
- consensus mechanism,
- event distribution,
- cryptographic services, and
- secure inter-node communications.

The DLT platform is described in more detail in 9.3.5.

5.4 DLT system interfaces

In general, interfaces can be used for communication by users, by administrators, between nodes in networks, to smart contracts, between smart contracts, between DLT systems, and to external non-DLT systems.

To understand the appropriate interfaces needed for interoperability, it is necessary to first understand which systems and/or applications are exchanging information and for what purpose.

Figure 2 illustrates four common kinds of interfaces used in DLT systems:

- Intersystem interfaces A – directly between two (or more) DLT systems;
- External interfaces B – between DLT systems and external non-DLT systems – like DLT oracles;
- User interfaces C – between user applications and DLT systems;
- Admin interfaces D – between admin applications and DLT systems.

Intersystem interfaces support communication between separate DLT systems.

External interfaces can provide a secure means to access capabilities outside the DLT system such as trusted data sources or functions. Such outside systems include off-ledger code, DLT oracles, non-DLT applications and off-ledger data. These are linked to a DLT node using external interfaces. (See 9.3.2 for an explanation of how DLT oracles work.)

A DLT system includes both the user applications providing end-user capabilities and the admin applications that provide capabilities for administration and management of the DLT system. These applications access the DLT system via the user interfaces and the admin interfaces of a node, respectively.

Smart contract interfaces can be needed between DLT services, so that smart contracts on one DLT system can interact with another DLT system. In addition, users and administrators might need to communicate with smart contracts.

Interfaces both impact and support interoperability – see 6.7 for interoperability.

5.5 Consensus

Consensus in the context of DLT systems, addresses the problem of agreeing on the content and the order of records in a widely distributed system where new records could be competing to be added by multiple nodes across this network.

DLT systems can operate in a potentially geographically dispersed network of nodes. Each node might be able to create a new record to be included into the ledger or receive information about a new record. But at that moment in time, that new record might not have been seen by all other nodes in the system. In some circumstances, a node might receive information about two different new records, both competing to be the most recent record. Consensus mechanisms figure out how all these independent nodes in the DLT system come to an agreement about the contents and order of these records.

There are different aspects of consensus, usually resolved over different timescales. First, there is the question of consensus of what transactions or sets thereof are valid. This issue is usually resolved during the creation of a DLT system, then is accepted by nodes when they join the network, and can be updated by governance mechanisms over the lifetime of the network. The initial and ongoing acceptance of the validation mechanisms is often established by kinds of informal social consensus (public blockchains or DLT systems) or by contractual means (private blockchains or DLT systems). Second, there is the question of which valid transactions should be included in the most recent ledger record (and by extension, all subsequent records).

Many different mechanisms can be used to achieve consensus about the inclusion and ordering of transactions. The choice of an appropriate mechanism can depend on the possible threat model or failure modes to be guarded against by the network of nodes. Some of these mechanisms are given below:

- Round Robin: Nodes from a small group take turns as the authority on the creation of new records, perhaps in a round-robin fashion.
- Byzantine Fault Tolerance or Byzantine Agreement: Conventional approaches within the distributed systems literature often consider the threat of “Byzantine failure”, where individual nodes in the network might not only be delayed in hearing new information from other nodes but might also be sent maliciously-constructed information from (a bounded number of) other malicious nodes. In this setting, a variety of “Byzantine fault-tolerant” consensus algorithms have been proposed, including Byzantine Paxos and PBFT. Normally, these require the number of nodes in the network to be known, for the number of malicious nodes to be small (typically, less than a third of the total number of nodes) and for every node to establish evidence from a quorum (typically, a clear majority) of other nodes. These kinds of mechanisms are often used in private DLT systems. In practice, good performance for these mechanisms might limit the size of the network to tens to hundreds of nodes.
- Nakamoto Consensus: For public blockchain systems, the participating nodes might be unknown, and their number can vary. Therefore, the nodes that should achieve consensus might be unknown. The consensus mechanism used in the Bitcoin blockchain, and in other public blockchains, such as

Ethereum, is known as "Nakamoto Consensus" and it addresses this challenge. The general scheme is that every node will accept as authoritative the longest sequence of blocks that it has seen. In the case where there are competing alternative blocks, there can be short-lived alternative histories ("forks", or "uncle blocks") temporarily accepted by various nodes within the network. However, it is likely that nodes eventually converge on acceptance of a common blockchain history, at least for older parts of the ledger. Some versions of consensus use not only the length, but also the total difficulty for Proof of Work, so the assessment can be multi-dimensional.

Proof of Work and Proof of Stake are two ways of mining a new block to accomplish Nakamoto Consensus:

— Proof of Work: The creation of new blocks in Bitcoin requires that a difficult cryptographic or computationally hard puzzle is solved: given a set of transactions for a new block, choosing a nonce that will make the hash-value for the block smaller than some currently-accepted difficulty target. Although finding a solution is hard, checking the solution is easy. The overall approach is called "Proof of Work". There is no known efficient solution to this kind of problem, and so a brute-force approach needs to be used. This means that the time required to find a solution is essentially random but varies in proportion to the computing power available to solve the problem.

  Proof of Work requires the investment of significant monetary capital into the computational resources needed to solve the problem. The nodes could receive an immediate return by being awarded both newly-minted cryptocurrency that can be claimed as a block reward, and transaction fees offered by individual transactions included in the block. This investment and return tend to align the incentives of the node with the creation of value and integrity in the overall DLT network. This scheme can accommodate any number of nodes in the network, by iteratively adjusting the puzzle difficulty over time in response to changes in the average time required for the network to solve each puzzle.

— Proof of Stake: In this mechanism, random leader election is determined by a kind of bet made in proportion to the amount of cryptocurrency staked by nodes competing to define the next block. The stake holding of nodes in cryptocurrency serves to align the incentives of the nodes with correct functioning of the network.

Other approaches for consensus are possible in permissionless DLT systems. For example, all nodes might periodically elect a small known number (tens to hundreds) of nodes to act on behalf of the whole network. Then, conventional consensus mechanisms that are normally appropriate for permissioned DLT systems can be used by the elected nodes. Given this, a reward system or incentive mechanism is a consideration (such as a payment) given to a party for undertaking some activity within the DLT system. An example of such activity is the work involved in running the consensus mechanism. Mining is an activity to seek rewards in some consensus mechanisms, and in those systems, a miner owns a node that runs mining programs.

Finality is a property of confirmed transactions that indicates whether they will remain part of the ledger after confirmation. Some DLT systems can have multiple non-identical blockchain histories. As the DLT network converges on a common blockchain history, the other copies of the blockchain history are discarded and as a result, some confirmed transactions might no longer be part of the ledger. This is particularly evident in DLT systems that use Nakamoto Consensus as the nodes will discard their current blockchain history for one that is longer. Thus, a confirmed transaction does not necessarily end up on the ledger. As external systems often depend upon a confirmed transaction being part of the ledger, a common practice is to wait until a certain number of blocks have been added to the ledger before concluding that the transaction will be part of the ledger. However even after waiting some number of blocks, there is still a possibility that some longer blockchain history will be uncovered that doesn't include the transaction. This type of finality is referred to as probabilistic finality and is in contrast to immediate finality used by DLT systems that cannot have multiple blockchain histories for a single ledger. With immediate finality, external systems know that once a transaction has been confirmed, it is guaranteed to be part of the ledger.