TECHNICAL REPORT

ISO/TR 7340

First edition
2023-01

Reference data distribution in financial services

Distribution de données de référence dans les services financiers

Reference number
ISO/TR 7340:2023(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 68, *Financial services*, Subcommittee SC 9, *Information exchange for financial services*.

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

0.1 Opening comments

With the increasing correlation between financial products, a lot of reference data (trading product, trading institution, trader information) are shared and reused in financial services. There is an urgent and significant worldwide demand for guidance and standardization of reference data distribution in financial services. Moreover, many industries expect efficient data distribution to ensure consistency, integrity, relevance and accuracy.

This document covers distribution modes (distributed and centralized), task scheduling, privacy protection, security and other issues. Data consistency and security are fundamental concerns for distributors, receivers, the ordered execution of the distribution tasks and independent distribution tasks of different receiver systems. Efficient distribution can achieve the goal of real-time synchronization of reference data, ensure that all organizations receive the most accurate data information in time and prevent system operation problems caused by information asymmetry.

This document's potential applications are independent of specific business scenarios and irrelevant to data type and data format specifications.

This document is intended to provide:

— reference information for distributors;

— new products and services for developers;

— benefits for receivers using reference data.

The purpose of this document is to simplify the data processing procedure, as well as improve the data distribution reliability and data sharing capabilities. Specifically, it will include two distribution modes: centralized distribution mode and distributed distribution mode. The former is traditional and the latter is emerging. Therefore, this document will be conducive to promoting new solutions for reference data distribution scenarios, such as distributed ledger technology. These benefits would be realized between certain service participants and within them.

0.2 How to approach this document

This document aims to provide a comprehensive insight into the development of reference data interfaces (RDIs) to realize efficient reference data distribution in financial services. In this sense, some aspects of the document are more mature than others. For example, the text is prescriptive where there is room to be so; where areas are less mature, commentary on good practice is provided and the considerations set out.

Broadly speaking, the document adopts the following logic:

— terms and definitions: all terms in the document;

— design principles: the principles and considerations for the design of the RDI;

— related technology: considerations and commentaries on different technologies;

— business model: the transmission process of public reference data and financial data standards;

— logical model: analysis of the logical structure of business data;

— physical model: overview and commentaries on the broker-based model and the non-broker-based model;

— interactions: considerations of the interactions between publishers and subscribers;

— QoS control: control of the network resource application in the transmission of reference data.

Reference data distribution in financial services

1 Scope

This document discusses the modes, related mainstream technologies, logical models, physical implementation models, data management (data storage and data security) and service quality control used in the reference data distribution in financial services.

This document applies to the reference data distribution and transmission processes in financial services.

2 Normative references

There are no normative references in this document.

3 Terms and definitions

For the purposes of this document, the following terms and definitions 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
reference data
shareable and reusable basic information in financial service scenarios

Note 1 to entry: A large amount of shareable and reusable basic information exists in financial service scenarios, such as legal entity identification codes (LEI), bank identification codes (BIC), bond issuers, buyers and sellers.

3.2
distributed ledger
data store through a network of distributed nodes

Note 1 to entry: Distributed ledger is a way of recording data that does not need to be stored or confirmed by any centralized entity.

Note 2 to entry: Distributed ledger is the most critical blockchain technology used in the capital market, an asset database that can be shared among multiple sites, different geographic locations or networks composed of multiple institutions.

3.3
financial technology
technology innovation of traditional financial products and services

Note 1 to entry: Financial technology uses various technological means to innovate the products and services provided in the traditional financial industry to improve efficiency and reduce operating costs.

3.4
full-duplex communication protocol
network protocol based on TCP

Note 1 to entry: Full-duplex communication protocol realizes full-duplex communication between the client and the server, which allows the server to send information to the client actively.

3.5
remote method invocation
Java interface class library

Note 1 to entry: Remote method invocation enables objects on the client-side virtual machine to call objects on the server-side Java virtual machine as if they were local objects.

3.6
FIX® protocol[1)]
Financial Information eXchange protocol
open electronic communications protocol designed to standardise and streamline electronic communications in the financial services industry, supporting multiple formats and types of communications between financial entities, including trade allocations, order submissions, order changes, executions reporting and advertisements

3.7
IMIX protocol
Inter-bank Market Information eXchange Protocol
financial industry standard based on the FIX protocol and widely used in the inter-bank market

3.8
RDI
reference data interface
set of well-defined methods, functions, protocols, routines or commands used for reference data

4 Principles

4.1 General

This clause covers the design principles that are considered up front when developing an RDI in financial services.

4.2 Compatibility

The RDI refers to some industry standards and is based on open architecture.

4.3 Data accuracy

Data accuracy is considered up front when developing the RDI to ensure that the data source can be monitored, the data can be transmitted in real time in batches and the data loss can be recovered.

4.4 High availability

High availability is considered to ensure no error accumulation and low data distribution latency to enable real-time communication.

4.5 Extensibility

Where possible, the RDI ecosystem is designed to be as extensible as possible to adapt to future use cases or scenarios. For example, software keeps an upgrade interface and upgrade space. In addition, the software entities (e.g. modules, classes, functions) are open for extension but closed for modification based on the open-closed principle.

1) FIX® is the trademark of FIX Protocol Limited. This information is given for the convenience of users of this document and does not constitute an endorsement by ISO of the product named. Equivalent products may be used if they can be shown to lead to the same results.

4.6 Security

The RDI ensures the security of user information and the information involved in the operation process. Furthermore, it repairs and handles various security vulnerabilities in a timely manner.

4.7 Maintainability

Maintainability includes code comprehensibility, testability, modifiability and system portability.

5 Related technology

5.1 Fintech

Financial technology (fintech) is a business model formed by the integration of finance and technology, specifically including digital payment, online lending, digital currency, equity crowdfunding and intelligent investment advisory. It mainly utilizes innovative technologies such as the internet, big data, cloud computing, blockchain and artificial intelligence to significantly affect the financial markets, financial institutions and the way financial services are provided.

5.2 WebSocket

The WebSocket protocol is a full-duplex communication protocol based on TCP. It implements full-duplex communication between the client and the server, allowing the server to send information to the client actively.

Most web applications implement long polling through frequent asynchronous JavaScript and XML (AJAX) requests, which is inefficient and wasteful of resources (because it requires constant connections, or the HTTP connection is always open). WebSocket abandons the traditional HTTP request/response mechanism and realizes a more flexible and accessible bilateral communication. The client browser first initiates an HTTP request to the server to establish a WebSocket connection. This request is different from the usual HTTP request as it contains some extra header information. One piece of additional header information called “Upgrade: WebSocket” indicates that this is an application for a protocol upgrade. The server side parses this additional header information and then generates a response message back to the client side. Finally, the connection is established and both parties transfer information freely through the channel until either the client or the server side actively closes the connection.

5.3 AJAX

Ajax (Asynchronous JavaScript and XML) is an integration of several technologies, including:

— dynamic display and interaction by DOM;

— data exchange and processing by XML and XSLT;

— asynchronous data reading by XML HTTP request;

— finally binding and processing data with JavaScript.

The principal of Ajax is an intermediate layer between the client and the server. Not all user requests are submitted to the server; the Ajax engine submits the request only when it is determined that new data needs to be read from the server. Through appropriate Ajax applications, some of the previous work of the server is transferred to the client. As a result, it can facilitate the processing on the client side and reduce the burden on the server and bandwidth.

5.4 RMI

RMI (Remote Method Invocation) is a core Java API class library that allows programs running on a Java virtual machine to access the objects running on different virtual machines (even if the different virtual machines are running on different physical hosts). RMI passes parameters to remote methods and returns results from remote methods calls.

5.5 Blockchain

Blockchain refers to a database distributed across locations (a distributed database) that acts as a digital ledger to record and manage transactions. Copies of the ledger are held by multiple parties themselves, data are added through negotiation by all parties and there is no need to have a third-party agent for managing the ledger.

The blockchain has the following characteristics:

— Immutable records: theoretically, the data added to the ledger is immutable and secure, and it disappears with the disappearance of the ledger; its content is jointly determined by all participants.

— No intermediaries: nodes can interact directly without intermediaries, which includes the ability for nodes to initiate data or digital asset transmission directly (perhaps a proprietary cryptocurrency, such as Bitcoin, or a digital representation of real-world assets, such as land ownership or fiat currency).

— No centralized controller: additions to ledger content or a change to the management structure are subject to negotiation by multiple participants.

— New opportunities to manage and share data: all participants can store and access data in various forms.

Therefore, blockchain can improve efficiency, trust and data identification for ledger holders.

While this technology is still in its development stage, it is clear that the blockchain has a lot of potential opportunities in many areas. In addition, standardization work related to blockchain is gradually being carried out and Technical Committee ISO/TC 307, *Blockchain and distributed ledger technologies*, has been established.

5.6 P2P

Communication between nodes in the P2P network is the process of data distribution. The tracker obtains the peer list and the node establishes connections with peers in the list based on the TCP protocol. The transmission between peers is realized by several messages, including 'keepalive', 'choke', 'un-choke', 'interested', 'not interested', 'bitfield', 'request', 'piece', 'cancel', 'have'. The 'keepalive' message is empty and used to guarantee the corresponding peer is online. The messages of 'choke', 'un-choke', 'interested', 'not interested' are responsible for notifying status information if updated. The other five messages are data messages accountable for transmission between nodes.

6 Business model

Reference data vendors distribute financial-related reference data products to financial market participants. These products vary in terminology, format, number of data elements and scope of coverage. Some proprietary and disparate reference data are supplied by the government, industry groups and private firms. Although these data are free to use, their proprietary nature runs counter to the publicly oriented consensus approach.

Financial market participants and vendors have led efforts to develop standards for financial identifiers and the underlying reference data describing financial instruments. With the development of technology, financial instruments are becoming more diverse and complex and continuously evolving. Every financial instrument represents a contract that governs the relationship between two or more

parties. Financial instrument reference data describe the terms and conditions of these contracts. The data vary in quality. Improving the quality poses unique challenges. Therefore, financial market participants and suppliers are committed to standardizing the reference data.

Figure 1 illustrates the transmission process of public reference data and financial data standards.

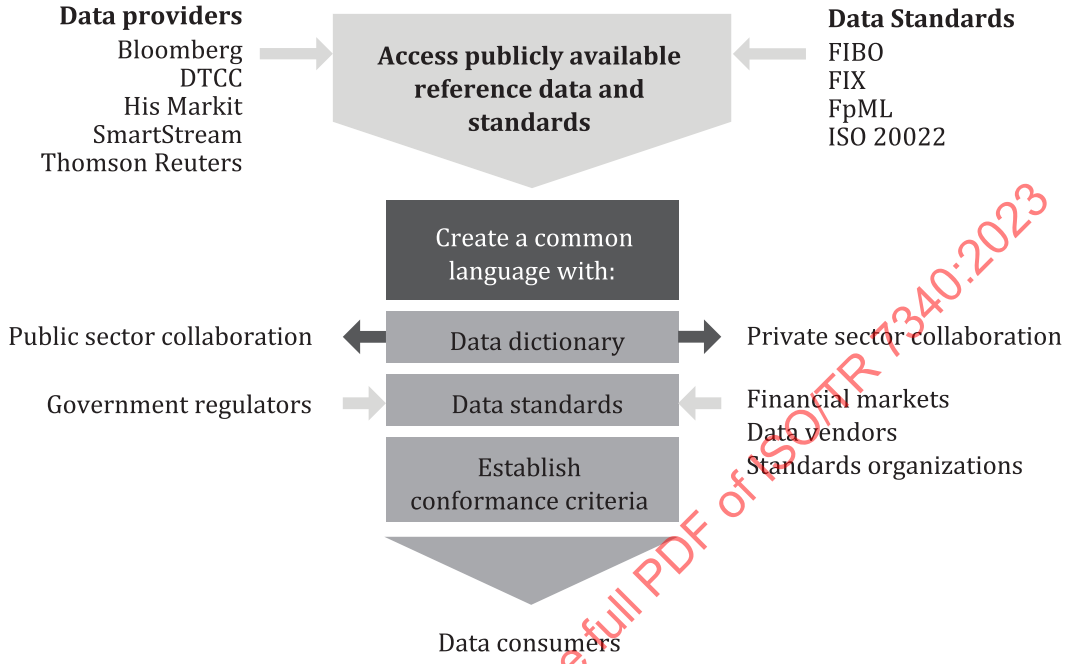

Figure 1 — Transfer process of public reference data and financial data standards

7 Logical model

7.1 Logical model

Publishers of reference data publish events, and subscribers subscribe to and receive events based on their interests. The main feature of publish/subscribe is the way notifications flow from senders to receivers. The receiver does not directly target a specific publisher but is indirectly addressed according to the content of the notification. Subscribers express their interested topics by notifying subscriptions on specific notifications, then asynchronously receiving messages that match their subscriptions, where the messages could be from any publishers who publish messages of corresponding topics.

Figure 2 illustrates the publishing and subscribing to the reference data process.

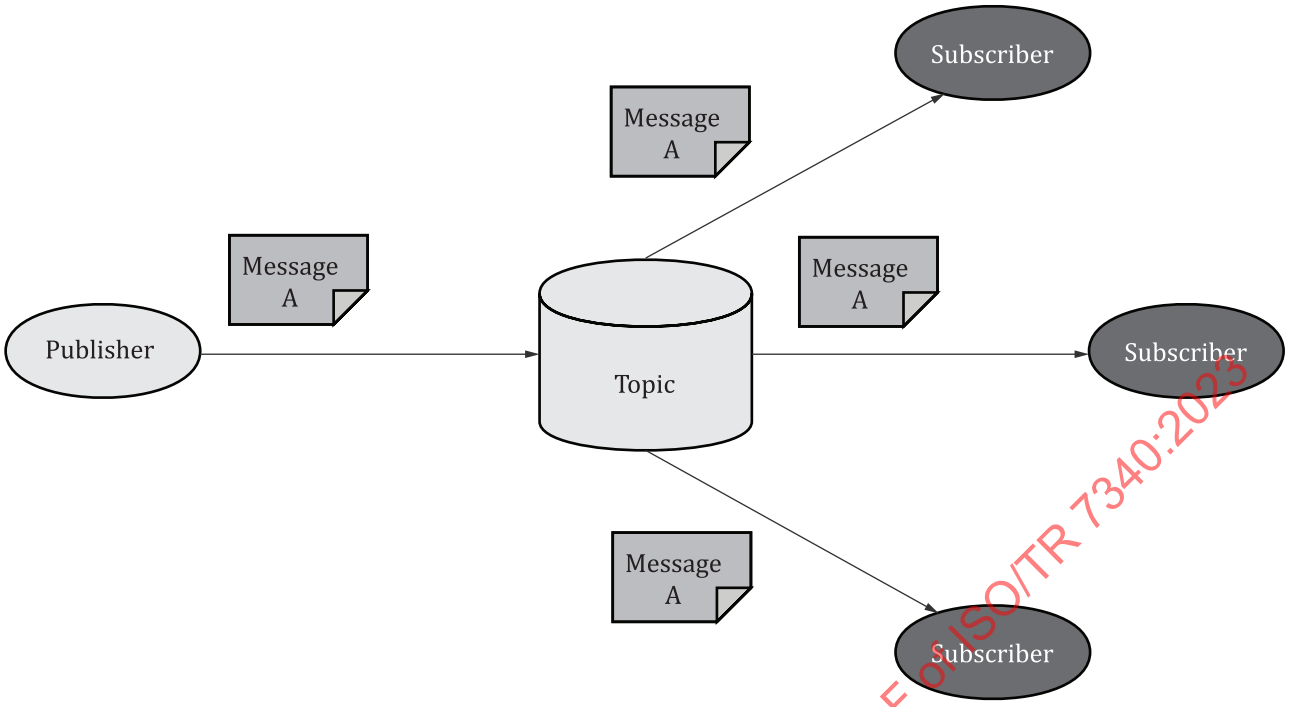

Figure 2 — Process of publishing and subscribing to the reference data

7.2 Distribution process

The reference data distribution process is divided into a publishing process and a subscription process. The publishing process includes five stages:

— initializing the participant objects;

— registering the published data types;

— creating the customized topic data information, the corresponding topic publishers and the related data writers;

— waiting for the corresponding topic subscribers;

— publishing related data.

The subscription process includes:

— initializing the participant objects;

— registering the published data types;

— creating the corresponding topics, the corresponding topic subscribers and the related data readers and listeners;

— implementing data reader monitoring.

7.3 Domains and topics

According to the relevant attributes of the data, the topic is divided into real-time type, policy-based type and best-effort type. Different types of data enter different data queues. When there is data to be distributed, the real-time type data are distributed first, followed by the policy-based type data and lastly the best-effort type data. This will enhance the quality of data distribution.

7.4 Publisher module

The publisher module mainly comprises the following classifiers:

— Publisher;

— DataWriter;

— PublisherListener;

— DataWriterListener.

The Publisher acts on behalf of one or several DataWriter objects that belong to it. When it is informed of a change to the data associated with one of its DataWriter objects, it will decide the appropriate time to send the updated message.

When making this decision, it will consider any extra information that goes with the data (e.g. timestamp, writer) as well as the QoS of the Publisher and the DataWriter.

All operations, except for the base-class operations set_qos, get_qos, set_listener, get_listener, enable, get_statuscondition, create_datawriter and delete_datawriter, possibly return the value NOT_ENABLED.

7.5 Subscription module

The subscription module comprises the following components:

— Subscriber;

— DataReader;

— DataSample;

— SampleInfo;

— SubscriberListener;

— DataReaderListener;

— ReadCondition;

— QueryCondition.

There are various subscription models based on how subscribers express their interest in a certain notification and how the matching is done by the notification system, so that the subscribers only receive information that they are interested in. Three publish/subscribe subscription models are presented: topic (subject)-based, content-based and type-based.

8 Physical model

8.1 Broker-based

Building an application-level network of brokers is the most common approach for designing a distributed notification service. Each broker communicates with its neighbour using TCP/IP, HTTP or IIOP/DCOM for subscription and publication. The main drawback of this approach is that the application-level topology possibly does not match the underlying physical network and the link between the two brokers is mapped to a “long” network path.

8.2 Non-broker

8.2.1 Multicast

The topic-based system mainly uses network-level multicast technology as each topic corresponds precisely to one multicast group. The most significant advantage of a network-level multicast lies in low latencies and high throughput. One of the drawbacks of this approach is that it cannot be easily deployed in wide area network (WAN).

8.2.2 P2P

Exploiting a peer-to-peer overlay network infrastructure for application-level multicasting, an overlay routing network is a logical application-level network built on top of a general network layer like an IP unicast. The nodes that are part of the overlay network can route messages between each other through the overlay network. There is an overhead associated with using a logical network for routing, as the logical topology does not necessarily mirror the physical topology. However, more sophisticated routing algorithms can be used and deployed, since routing is implemented at the application level.

The most recent development in P2P technology is the blockchain. In the blockchain, the ledgers for transactions are maintained over public P2P networks, which is more reliable than other P2P implementations.

8.3 Interactions

8.3.1 General

Interactive data are divided into the request data and the response data. The client initiates a request to the server by the request data using the request subject. The server returns a response to the client by the reply data using the reply subject.

8.3.2 Publisher view

In the publisher's view, the response data of the server consists of a header part and a data part. The header part includes a globally unique identifier (GUID) of the client that initiated the request and the serial number of the request corresponding to the response. The content of the data part is the return information of the function.

8.3.3 Subscriber view

In the subscriber's view, the request data of the client also includes a header part and a data part. The header part includes a GUID, a bound server name, a called service name and function, a request serial number, and so on. The content of the data part is the parameter required for the function call.

8.3.4 Smart contract

The smart contract technology is deterministic, consistent, verifiable and decentralized. The same input on different computers or on the same computer at different times will produce the same output through smart contracts. Through digital signature technology, the contract is traceable and unchangeable. All nodes in the contract can get the status and operation records, and the execution process is verifiable. The decentralization of blockchain makes it impossible for any node in the contract to unilaterally modify the content, which greatly reduces the risk of human intervention.

8.3.5 Accuracy

The computer network architecture is divided into the following seven layers:

— Physical layer: converts data into electronic signals that can be transmitted over physical media.

— Data link layer: determines how to access network media. Data are framed, flow control is handled at this layer, and this layer also specifies the topology and provides hardware addressing.

— Network layer: arranges data routing in large networks.

— Transport layer: provides reliable end-to-end connections.

— Session layer: allows clients to use easy-to-remember names to establish connections.

— Presentation layer: negotiates the data exchange format.

— Application layer: an interface between the client's application and the network.

The data transmission goes through these seven layers of interaction. The data error at each layer will lead to inaccurate reference data being received by the final subscriber. For example, the application layer cannot guarantee hardware service efficiency or network service quality. When the publisher sends a large file to the subscriber which has to be separated into several sub-transmissions, the publisher is faced with the issue of how to ensure that the subscriber finally receives complete and accurate data, and the data are not lost or tampered with during the transmission. To solve this problem, it adds a check code to the transmitted data, which is used to ensure the accuracy of data reception. The method of generating the check code is only known by the data sender and the data receiver. For example, the method can be an md5 code generated by adding a constant string to the transmitted data. The data sender generates the md5 code and sends it to the receiver together with the raw data. Once the raw data are received by the receiver, they can use the same algorithm to generate an md5 code and compare it with the received one. The received data are complete and accurate only when the two md5 codes are consistent.

9 Data payload syntax

9.1 General

The data format and syntax structure are compatible with International Standards, and when developing the RDI, the current mainstream data structures are used, including ISO 20022 (data object standard for transmitting financial information) and FIX (an open financial information transmission protocol provided by FIX Association).

As the data structures in the RDI follow ISO 20022 and FIX, the generated reference data are coded under the ISO 20022 schema and the FIX schema, respectively.

9.2 Syntax and structures

The physical level delivers the physical implementation of MessageDefinitions and constraints in an appropriate syntax, such as ISO 20022 XML. Specific design rules are used to build the physical representation of the MessageDefinitions from the deliverables of the logical level.

The key deliverable of the physical level is:

— a set of ISO 20022 syntax message schemes;

— the message orchestration.

9.3 Data types

The objective of a DataType is to unambiguously specify the value space of a BusinessAttribute, a MessageBuildingBlock or a MessageAttribute. XSD's built-in DataTypes can be used directly as types of BusinessAttribute, MessageBuildingBlock and MessageAttributes where, in contrast to the user-defined DataTypes, they cannot be further constrained.

Each datatype imported from the XSD built-in library represents a set of values (i.e. its value space). The DataType metaclass represents the collection of all value sets without identity. Therefore, DataTypes imported from XSD built-in DataTypes library are the instances of the DataType metaclass, which do not provide values for the metaproperties.

10 Authority

10.1 Access control

This subclause describes some access control techniques used in distributed messaging systems, for example how it is ensured that consumers do not receive irrelevant messages in a content-based publish/subscribe system when messages are sent from publishers to subscribers through multiple brokers. Several changes to the original access control policy of the messaging system are proposed, such as access control version identifiers.

10.2 Privacy protection

The flexibility of the publish/subscribe system comes with increasing challenges and costs in data security and privacy. In addition to the traditional data security concerns, such as the confidentiality and integrity of messages, the authentication of the source, access control and authorization of subscribers, publish/subscribe also poses new challenges to message distribution schemes. In classic layered communication systems, data in the application layer can be protected by various security mechanisms such as encryption and message authentication without affecting the underlying data distribution mechanisms implemented in the network layer. In the case of the publish/subscribe system, content protection with similar security mechanisms would conflict with its distribution function, since the latter relies on the content it transmits. Therefore, in terms of privacy protection, it expects new solutions to allow intermediate nodes to perform routing operations with those data encryption and integrity mechanisms.

First, a secure distribution mechanism is established, which uses encrypted content as a search key to implement look-up in the distribution table. Furthermore, a big concern in the content-based publish/subscribe system is how to ensure the confidentiality of the subscriber's messages when they register their interests to the server. While encrypting these messages appears to be a suitable solution for user privacy protection, such encryption operation raises additional challenges to the distribution mechanism. Hence, in addition to the search key for the look-up mechanism, the forwarding table itself would be based on encrypted data as well.

11 Security

11.1 Data security

Encryption refers to converting plain text into secure code that can only be deciphered with a decryption key. This ensures that the run-time data across the network and the web, as well as the static data in the cloud or data centre, cannot be seen by anyone without the key (even if it is stolen), adding a solid layer of security.

The use of data encryption in financial services firms has several regulatory guidelines, including the Federal Financial Institutions Examination Council (FFIEC) and the new General Data Protection Regulation (GDPR). This subclause discusses the application of data encryption during the data distribution process.

11.2 Transport security

The document follows the guidelines for Secure Use of Transport Layer Security in BCP195,[4] such as:

— TLS version 1.2[5] or later;

— a TLS server certificate check as per RFC6125.[6]

The implementation of these standards will ensure:

— the confidentiality of the data sent over the connection, as the connection is encrypted;

— the integrity of the data sent over the connection, as it is not possible for an attacker to inject or tamper with the data.

In addition, when a trusted certificate authority issues the server certificate used in the TLS handshake, the client has a strong assurance of the identity of the server it is interacting with.

More specifically, when the server wishes to authenticate the client at the transport layer and mutual TLS authentication is implemented, the client will present its certificate to the server as part of the TLS handshake and give the server a strong assurance as to the identity of the client. This is a common practice for many existing financial API implementations, and it is implemented for developing the RDI.

11.3 Application security

As detailed in RFC6749,[7] in the traditional client-server authentication model the client software requests an access-restricted resource (protected resource) on the server by authenticating with the server using credentials issued by the server. This could be the user name and password of an individual user or an API key granted to a company.

Using such credentials directly via an API creates several problems and limitations, especially in cases where a user wants to allow third-party client software to access their resources:

— third-party applications will store the user's credentials for future use, typically a password in plain-text;

— servers support password authentication, despite the password itself having security flaws;

— third-party applications gain overly broad access to the user's protected resources, leaving users without any ability to restrict duration or access to a limited subset of resources;

— users cannot revoke access to an individual third party without revoking access to all third parties;

— compromises of any third-party application will result in compromises of the end-user's password and all of the data protected by that password.

12 QoS control

12.1 General

QoS is a set of technologies that cost-effectively manage network traffic to enhance user experiences in home and enterprise environments. Real-time applications focus on the predictable transfer of data with minimal cost. Modifying the QoS of an entity can specify the behaviour of the entity at runtime. The developers can only focus on setting QoS.

A large number of real-time applications put their communication patterns model into a purely data-centric exchange model, in which one application is responsible for publishing data and the other for accessing the data they are interested in. The main concern of these real-time applications is how to achieve predictability of the data distribution with the minimum cost. The demand is transformed into QoS requirements, which will affect the predictability, overhead and resource usage of real-time applications.

This document describes several QoS policies. These QoS policies describe constraints on service behaviour. The appropriate policies can satisfy the flexible exchange of large amounts of complex data, enabling corresponding middleware operations, such as time management and data consistency.