INTERNATIONAL STANDARD

ISO 5231

First edition
2022-10

Extended farm management information systems data interface (EFDI) — Concept and guidelines

Reference number
ISO 5231:2022(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 23, *Tractors and machinery for agriculture and forestry*, Subcommittee SC 19, *Agricultural electronics.*

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

Participants in the agricultural environment are dealing with the necessity of exchanging data with various systems and interfaces. With increasing demand for process monitoring and control, and just-in-time information on task execution state, and at the same time integration of mobile devices into farm work processes, the need of a standardized way for data exchange arises.

Extended farm management information systems data interface (EFDI) — Concept and guidelines

1 Scope

This document specifies an extensible communication system concept and defines rules for adding new functionalities to cover specific use cases.

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
network
group of two or more participants connected to each other via a server

3.2
endpoint
uniquely addressable instance within a network

Note 1 to entry: An endpoint can be a farm management information system (FMIS), a telemetry unit, a terminal or a complete machine.

Note 2 to entry: The server is also an endpoint.

3.3
client
C
endpoint that communicates with the server

3.4
server
S
central component for communication in a network

Note 1 to entry: All communication is done via the server.

3.5
messaging component
MC
part of the server and network management

Note 1 to entry: The messaging component (MC) keeps track of logged-in endpoints and their capabilities and is also responsible for the delivery and forwarding of messages.

3.6
noun
type of a message

3.7
verb
action which is executed with a specific noun

3.8
scenario
order in which the request and response messages shall be performed

Note 1 to entry: Every request and response consists of a verb and a noun.

3.9
scenario flow
sequence of scenarios

Note 1 to entry: Scenario Flows define how scenarios are related to each other and how they are to be executed in dependence on each other.

3.10
scenario set
group of scenario flows

3.11
A-Step
request that is sent from client A to client B

3.12
B-Step
request that is sent from client B to client A

3.13
streaming
subscription to specific types of messages and subsequently reception of unsolicited information

3.14
onboarding
initial access or registration of an endpoint

3.15
onboarding service
service enabling clients to access a network

3.16
protocol buffers
protobuf
data structures that can be serialized in compact binary representation

Note 1 to entry: In this document protocol buffers version 3 (proto3) applies to all definitions. See also https://developers.google.com/protocol-buffers/docs/proto3

3.17
message queue telemetry transport
MQTT
standardized transport protocol with publish-subscribe paradigm

Note 1 to entry: In this document MQTT 3.1.1 applies to all definitions. See also http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.html

3.18
hypertext transport protocol secure
HTTPS
set of rules for secure transferring data over the internet or a local network

Note 1 to entry: See also https://datatracker.ietf.org/doc/html/rfc2818

3.19
semantic versioning
concept for defining version numbers to software.

Note 1 to entry: Semantic versioning allows to imply compatibility information to a version number, see also https://semver.org/.

3.20
VDMA Database
contains all ISO5231 definitions for scenario sets, scenario flows, scenarios and message.

Note 1 to entry: Database can be accessed at https://isobus.net/isobus/efdi

3.21
transport layer security
TLS
protocol for secure communication over the internet

3.22
transport layer security certificates
TLS-CERTS
certificates providing information for encryption of communication

3.23
extensible messaging and presence protocol
XMPP
application profile for data exchange

3.24
fully qualified domain name
FQDN
location with all its domain levels

4 Architectural overview

The extended farm management data interface consists of a set of layers packaged on top of the network application layer as illustrated in Figure 1.

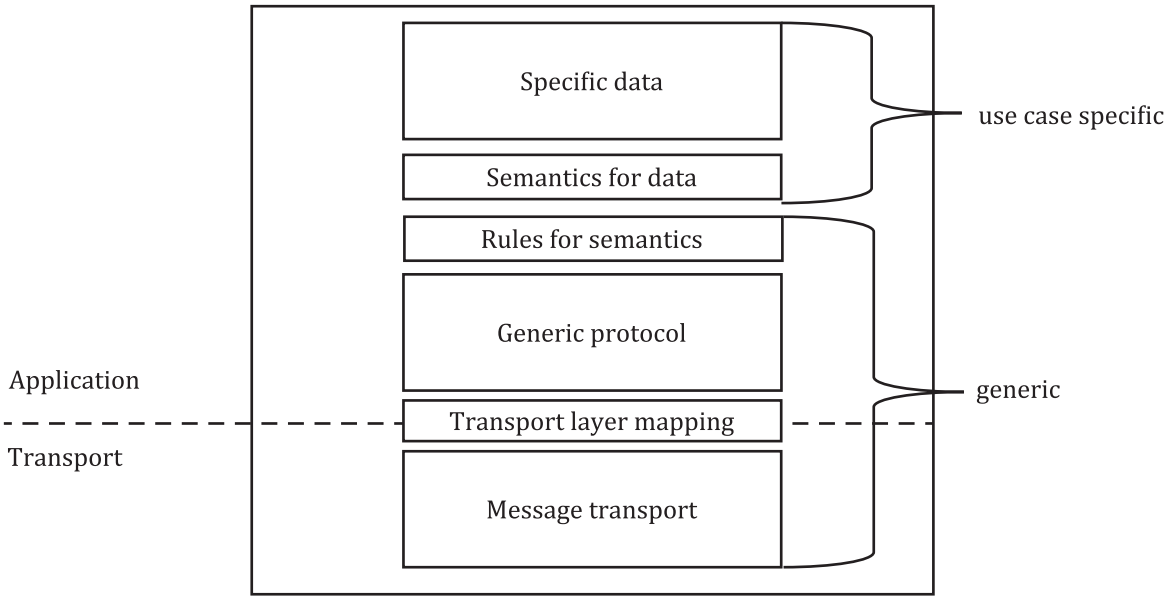

Figure 1 — Layer model

The selected message transport protocol to show how to implement EFDI is based on the MQTT protocol as the message transport layer. However, the application layers (see Figure 1) are not bound to it. Messages are partitioned into a header and a payload. The header shall contain addressing information and may contain additional data for setting up a stateful, persistent session. Depending upon lower-layer transport protocol functionalities, the additional data may be omitted and instead be covered by the transport protocol layer. It is theoretically possible to transport any payload within the EFDI protocol. However, transfer of ISO 11783-10 data is explicitly described within Annex B. There is also a generic file transfer handling described by the basic file functionality scenario set. Apart from the use case specific layer sitting on the top-most position of the stack, EFDI provides a semantic layer to indicate to message receivers what to do with the data (execute task, replace element within task, etc.).

The definition of generic protocol layer is described in Clause 6. The rules for semantics are defined in Clause 7 and the transport layer mapping is described In Clause 8.

5 Network

5.1 Overview

This clause gives an overview about the different components that are part of the network as shown in Figure 2.

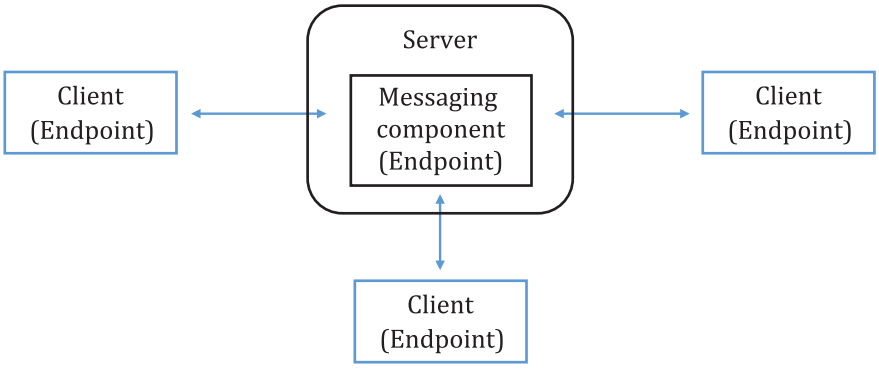

Figure 2 — Components in a network

Clients can establish a connection to the server. All endpoints in the network can communicate to each other.

5.2 Onboarding service

The onboarding service (OS) is part of the server component of the network. It is a service with which an endpoint can connect to the network. Individual IDs and certificates are generated, which are necessary for subsequent communication. The onboarding is part of the registration process of an endpoint.

5.3 Connected networks

Interconnecting multiple networks can be achieved by implementing a server containing both a messaging component and a client.

In Figure 3, server A handles the directly connected clients via its messaging component. Additionally, server A connects as a client to server B. Through this connection server A indirectly connects to clients through server B. Depending on the provided capabilities of server A's client, server B may also be able to indirectly connect to server A's clients.

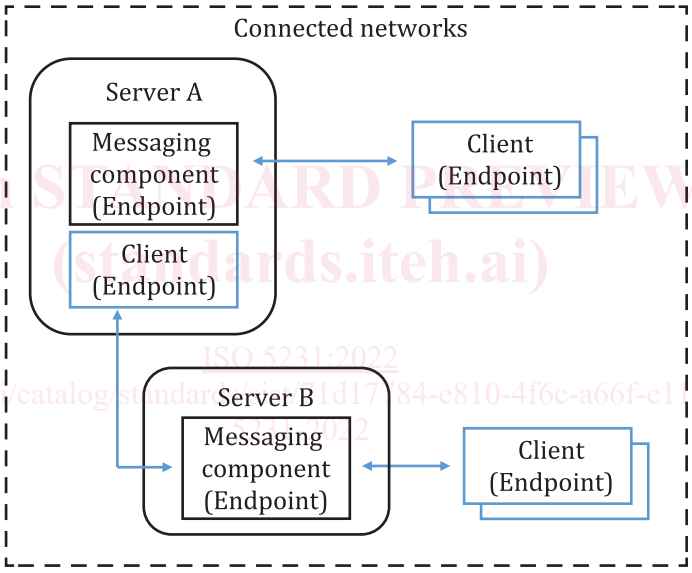

Figure 3 — Components in connected networks

6 Messaging

6.1 Overview

This clause presents various basic concepts of how communication takes place. It includes description of the construction of messages, architectural structures, addressing and streaming.

6.2 Hierarchical structure of scenario sets, scenario flows and scenarios

6.2.1 General

The hierarchical structure of scenario sets, scenario flows and scenarios are depicted in Figure 4. Generally, the scenario set is the topmost item, which contains at least one scenario flow. Each scenario flow contains at least one scenario. A scenario consists of a request and a response. Both a request and a response consist of a combination of a verb and a noun.

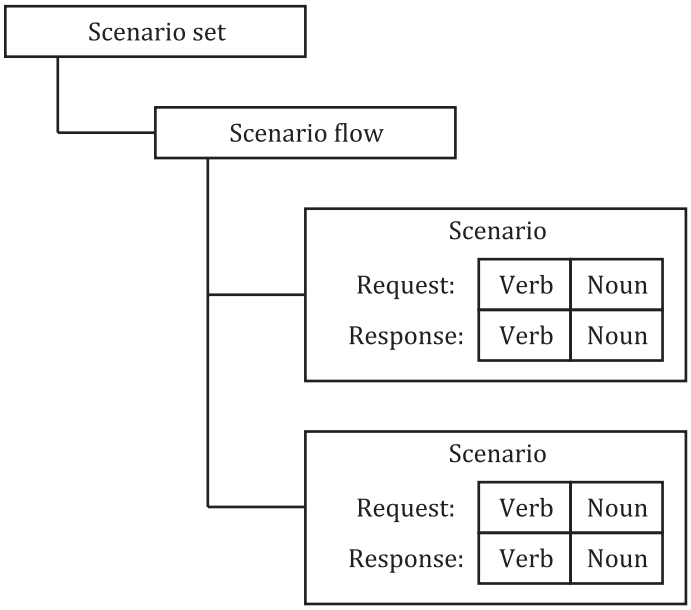

Figure 4 — Hierarchical structure of scenario sets, scenario flows and scenarios

All of the described messaging follows a common process. This common process is described in 6.2.2 to 6.2.5.

6.2.2 Scenario

A sequence of request and response messages is called a scenario. All of the scenarios have a request-response messaging pattern. That means that there is a defined response for a specific request. The request is sent by an endpoint and the response is sent back by another endpoint.

The requests and responses are defined as a combination of verb and noun. This verb-and-noun concept is based on the Business Object Document specified by the Open Applications Group Integration Specification (OAGIS), see Annex A for additional information. The verb defines the action or actions that are desired for the included business information. The business information itself is contained in the other sub-section called a noun.

An example is a scenario for requesting the list of endpoints that are connected to the server shown in Figure 5. The request contains the verb GET and the noun ListEndpoints. Translated this means that the list of connected endpoints is requested. The response contains the verb SHOW and the noun ListEndpointsResponse. Translated, this means that the list of connected endpoints is returned.

Figure 5 — ListEndpoints scenario

The schematic representation can be seen in Figure 6. Additional information and how scenarios shall be described formally can be found in 8.2.

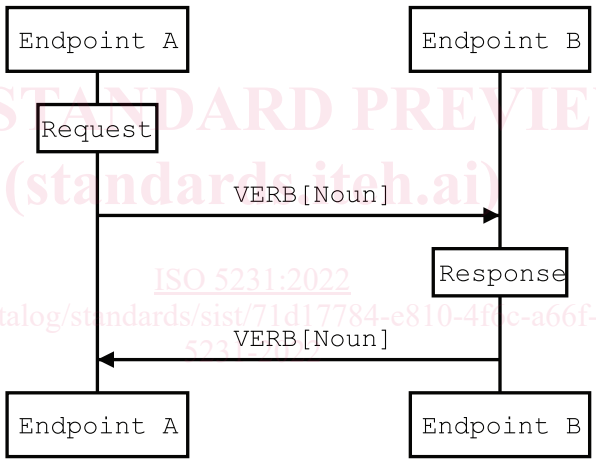

Figure 6 — Schematic representation of scenario

The list of available verbs based on the OAGIS standard has been extended in order to cover new use cases. The following verbs have been added:

— FORWARD: shall only be used by the server, indicates that a message was received and will be forwarded to other endpoints. A more detailed explanation and when this verb is used can be found in 6.2.5;

— STREAM_START: to request streaming data, indicates that the endpoint would like to receive streamed data from now on;

— STREAM_STOP: indicates that the endpoint no longer wishes to receive streamed data;

— STREAM_DATA: necessary to mark streamed data.

More detailed information about the new verbs for streamed data can be found in 7.6.