SLOVENSKI STANDARD
SIST EN 50631-4-1:2023

01-maj-2023

Omrežje gospodinjskih aparatov in povezljivost mreže - 4-1. del: Posebni vidiki komunikacijskih protokolov: SPINE, SPINE-IoT in SHIP

Household appliances network and grid connectivity - Part 4-1: Communication Protocol Specific Aspects: SPINE, SPINE-IoT and SHIP

Netzwerk- und Stromnetz-Konnektivität von Haushaltsgeräten - Teil 4-1: Spezifische Aspekte der Kommunikationsprotokolle: SPINE, SPINE-IoT und SHIP

Appareils domestiques connectés au réseau et réseau intelligent - Partie 4-1: Aspects spécifiques des protocoles de communication: SPINE, SPINE-IoT et SHIP

Ta slovenski standard je istoveten z: **EN 50631-4-1:2023**

ICS:

97.120	Avtomatske krmilne naprave za dom	Automatic controls for household use

SIST EN 50631-4-1:2023 **en**

EUROPEAN STANDARD

NORME EUROPÉENNE

EUROPÄISCHE NORM

EN 50631-4-1

March 2023

ICS 97.120

English Version

Household appliances network and grid connectivity - Part 4-1: Communication Protocol Specific Aspects: SPINE, SPINE-IoT and SHIP

Appareils domestiques connectés au réseau et réseau intelligent - Partie 4-1: Aspects spécifiques des protocoles de communication: SPINE, SPINE-IoT et SHIP

Netzwerk- und Stromnetz-Konnektivität von Haushaltsgeräten - Teil 4-1: Spezifische Aspekte der Kommunikationsprotokolle: SPINE, SPINE-IoT und SHIP

This European Standard was approved by CENELEC on 2023-02-13. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CENELEC member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CENELEC member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Bulgaria, Croatia, Cyprus, the Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, the Netherlands, Norway, Poland, Portugal, Republic of North Macedonia, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Türkiye and the United Kingdom.

CENELEC

European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

CEN-CENELEC Management Centre: Rue de la Science 23, B-1040 Brussels

Contents

Page

European foreword

This document (EN 50631-4-1:2023) has been prepared by WG 07 “Smart Household Appliances” of CLC/TC 59X “Performance of household and similar electrical appliances”.

The following dates are fixed:

- latest date by which this document has to be implemented at national level by publication of an identical national standard or by endorsement (dop) 2024-02-13
- latest date by which the national standards conflicting with this document have to be withdrawn (dow) 2026-02-13

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CENELEC shall not be held responsible for identifying any or all such patent rights.

Any feedback and questions on this document should be directed to the users’ national committee. A complete listing of these bodies can be found on the CENELEC website.

Introduction

Energy management systems will more and more become necessary due to change from fossil and nuclear to renewable production and the associated decentralization. Since an appropriate standard for a home and building management is in preparation this document specifies how sets of products from multiple manufacturers can exchange information with Home and Building / Customer Energy Management Systems, located in a home network or in the cloud.

This document focuses on interoperability of household appliances and describes the necessary control and monitoring. It defines a set of functions of household and similar electrical appliances. The functions in this document cover next to energy-management main remote-control and – monitoring use cases.

This document does not deal with safety and security requirements. Safety requirements have been set in the IEC/EN 60335 series [1].

EN 50631 will provide interoperability on information exchange among various appliances in the home. The EN 50631 series will be re-arranged regarding the further development and will be split into 6 parts:

— EN 50631-1, *Household appliances network and grid connectivity — Part 1: General Requirements, Generic Data Modelling and Neutral Messages*

— EN 50631-2, *Household appliances network and grid connectivity — Part 2: Product Specific mappings, details, requirements and deviations*

— EN 50631-**3-x**, *Household appliances network and grid connectivity — Part 3: Specific Data Model Mapping*

— EN 50631-**4-x**, *Household appliances network and grid connectivity — Part 4: Communication Protocol Specific Aspects*

— EN 50631-5, *Household appliances network and grid connectivity — Part 5: General Test-Requirements and - Specification*

— EN 50631-6, *Household appliances network and grid connectivity — Part 6: SPINE Data Model Toolbox*

Data communication heavily depends on the environment of appliances. Sometimes low bitrate or energy efficient communication puts strict requirements to selected communication technologies. Therefore, popular and de facto standards had been and will be developed by the industry to fulfil such requirements. To not influence common data modelling for appliances because of such restrictions, the standardized data models and neutral message structures need to be applied to communication technologies.

This standard series therefore is intended to separate data modelling and neutral message structure from the attached communication.

Part 1 defines general requirements, generic data modelling and generic neutral messages without relation to any specific communication technology or any product specific layout.

Part 2 lists and specifies product specific requirements and implementation guidance based on the generic data model and generic neutral messages.

Part 3 defines the mapping of neutral messages to examples of typical data models like SPINE, OCF, and so forth. These data models are neither mandatory nor to be seen as complete spectrum of data models.

Part 4 defines the mapping of neutral messages to examples of typical communication protocols. These communication protocols are neither mandatory, nor do they provide an exhaustive list of communication protocols.

Part 5 defines testing requirements and testing specifications. This part will be covered in the future by a New Work Item Proposal.

Part 6 provides the technical reference specification for the SPINE data model. This part will be covered in the future by a New Work Item Proposal.

1 Scope

This document specifies the application of relevant transport protocols for Home and Wide Area Networks as well as cloud connectivity; in this case, SPINE (Smart Premises Interoperable Neutral-Message Exchange), SPINE-IoT, and SHIP (Smart Home IP).

This document is part of the EN 50631 series, which defines the information exchange between Smart Appliances and management systems in homes and buildings including energy management.

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.

IETF RFC 793:1981, *Transmission Control Protocol*

IETF RFC 3280:2002, *Internet X.509 Public Key Infrastructure Certificate Revocation List (CRL) Profile*

IETF RFC 6455:2011, *The WebSocket Protocol*

IETF RFC 6763, *DNS-Based Service Discovery*

3 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

ISO and IEC maintain terminological 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
CA
Certificate Authority
Certification Authority
entity which can provide a digital signature for certificates

Note 1 to entry: Other SHIP nodes can check this digital signature with the certificate from the CA itself, the “CA-certificate”.

3.2
Commissioning Tool
<SHIP> instrument to establish the trust between different devices in the smart home installation, e.g. distribute trustworthy credentials from some SHIP nodes to other SHIP nodes

Note 1 to entry: E.g. a smart phone, a web server or a dedicated device can embody the role of a commissioning tool. So far, the SHIP specification does not specify a commissioning tool; an interoperable protocol for commissioning can be used on the layer above SHIP.

Note 2 to entry: A manufacturer may also use their own solutions.

3.3
DNS
Domain Name System

[SOURCE: IETF RFC 1035]

3.4
DNS host name
fully qualified domain name used within DNS as host name to get the IP address of the corresponding internet host

3.5
DNS-SD
Domain Name System – Service discovery

[SOURCE: IETF RFC 6763]

3.6
EUI
Extended Unique Identifier

Note 1 to entry: https://standards.ieee.org/develop/regauth/tut/eui64.pdf

3.7
factory default
setting that allows the user to reset the SHIP node to the as-new condition

Note 1 to entry: This means that all data that has been provided and stored by the SHIP node during its operation time SHALL be deleted.

3.8
IANA
Internet Assigned Numbers Authority

3.9
IP
Internet Protocol

3.10
LAN
Local Area Network

3.11
MAC
Media Access Control

3.12
mDNS
multicast DNS host name
fully qualified domain name used within mDNS as host name to get the IP address of the corresponding local SHIP node

3.13
M/O/NV/C
abbreviations which refer to:

1. M = mandatory
2. O = optional
3. NV = not valid
4. C = choice, i.e. a presence or support depends also on the selection from multiple possibilities

and which are primarily used within specific definition tables describing certain specialized data model definitions

3.14
numerical representation
written system for expressing numbers

EXAMPLE 0xab represents a decimal value of 171

3.15
PIN
Personal Identification Number
specification which makes use of a PIN as secret for SHIP specific verification procedures

3.16
PKI
Public Key Infrastructure

3.17
Push Button
switching mechanism to control some aspects of a machine or a process

Note 1 to entry: A push button event does not necessarily mean that a real physical button has to be used to trigger this event. A push button event may also be generated by other means, e.g. via a smart phone application or a web-interface (secure connection to SHIP node required). A push button shall provide a simple mechanism for a user to bring the device to a certain state or start a certain process.

3.18
QR Code
registered trademark of DENSO WAVE INCORPORATED which is the short form for “Quick Response Code” and used for efficient encoding of data into a small graphic

Note 1 to entry: Among others, international standard ISO/IEC 18004:2015 specifies the encoding of QR code symbols.

3.19
RFC
request for comments

3.20
SHIP
Smart Home Internet Protocol

Note 1 to entry: This term is used throughout this document to refer to the described communication protocol.

3.21
SHIP ID
identification which is used to uniquely identify a SHIP node, e.g. in its service discovery, and which is present in the mDNS/DNS-SD local service discovery

Note 1 to entry: Each SHIP node has a globally unique SHIP ID.

Note 2 to entry: See 6.5.

3.22
SHIP Client
role which is assigned to the SHIP node that also embodies the TCP client role for a specific peer-to-peer connection

3.23
SHIP Commissioning
<SHIP> term which denotes the distribution of trustworthy SKIs from certain SHIP nodes to other SHIP nodes

Note 1 to entry: The distribution of the trustworthy SKIs is handled by a so-called SHIP commissioning tool.

3.24
SHIP Commissioning Tool
instrument which is used to distribute trustworthy SKIs from certain SHIP nodes to other SHIP nodes, and which allows a user to handle the trust relationships in the whole SHIP installation (if each node in the installation supports commissioning) over one simple user interface

3.25
SHIP Node
logical device which communicates via the described SHIP protocobe and can be integrated into a web server or physical device

Note 1 to entry: One physical device may have more than one logical SHIP node. In this case, each SHIP node MUST use distinct ports (e.g. a physical device provides 3 open ports with 3 different SHIP services).

3.26
SHIP Server
role which shall be assigned to the SHIP node that also embodies the TCP server role for a specific peer-to-peer connection

3.27
SKI
Subject Key Identifier
identifying certificates that contain a specific public key and is used as a cryptographically backed identification and authentication criterion

Note 1 to entry: Each SHIP node has a specific public key.

3.28
SPINE
Smart Premises Interoperable Neutral message Exchange
Technical Specification of EEBus Initiative e.V

3.29
Trusted SHIP Node
term only applicable from a specific SHIP node point of view

Note 1 to entry: If SHIP node A has a communication partner and a trusted relationship to SHIP node B, SHIP node B is called a trusted SHIP node from SHIP node A's point of view; a trusted relationship can be established in different ways.

3.30
UCS
Universal Character Set

3.31
UTF
UCS Transformation Format
computing industry standard for the consistent encoding, representation, and handling of text expressed in most of the world's writing systems

3.32
WAN
Wide Area Network

3.33
web server based SHIP node
SHIP node that is hosted by a web server

3.34
WiFi
IP networks based on the IEEE 802.11 set of standards, used for wireless IP communication

4 SPINE-IoT Protocol

4.1 General

This document specifies the application of relevant transport protocols for Home and Wide Area Networks as well as cloud connectivity; in this case, SPINE (Smart Premises Interoperable Neutral-Message Exchange), SPINE-IoT, and SHIP (Smart Home IP).

Figure 1 shows the use of the transport protocols within this document. In case of local operation (the Smart Appliance is connected to a local Customer Connectivity Manager via the HAN), the transport protocols SPINE and SHIP SHALL be used. In case of IoT/Cloud operation (the Smart Appliance communicates via its cloud representation or directly with an IoT/cloud based Customer Connectivity Manager), the transport protocol SPINE-IoT via HTTP/ REST API / Open API SHALL be used.

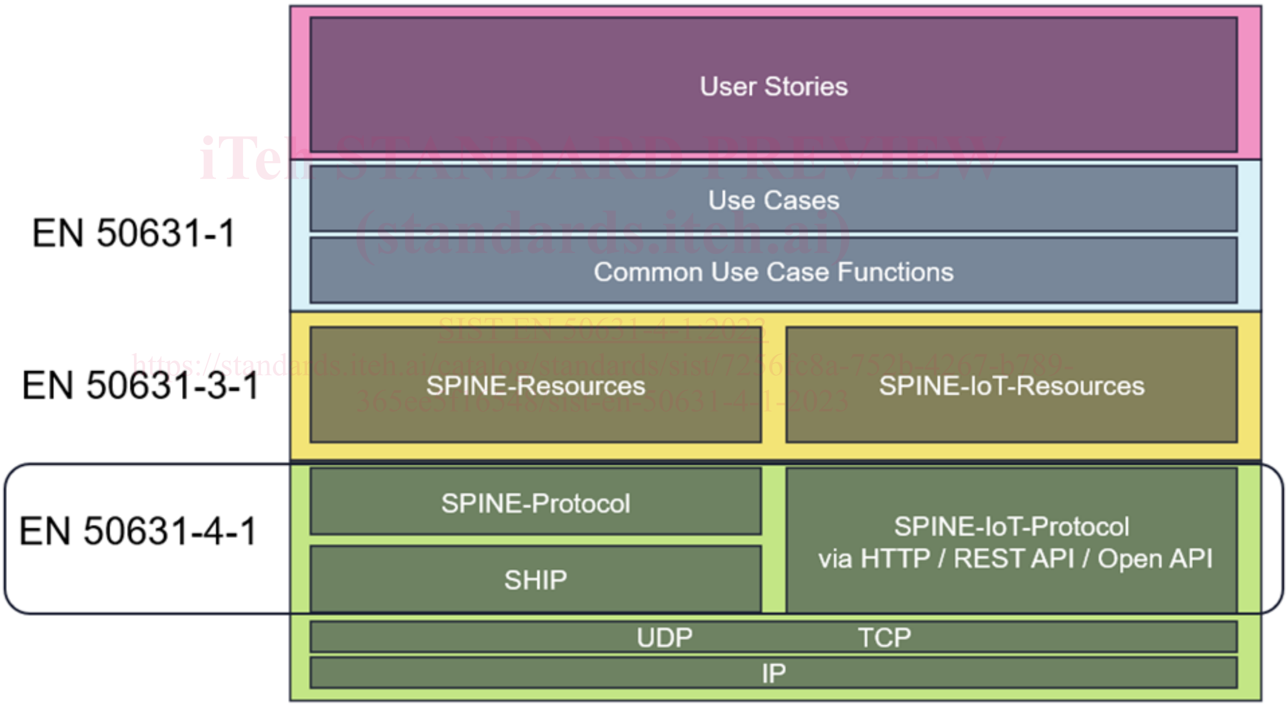

Figure 1 — Overview of transport protocols within EN 50631-4-1

SPINE-IoT (Smart Premises Interoperable Neutral-message Exchange for Internet of Things) defines a neutral IP-based layer which helps connecting different communications technologies (i.e. Wi-Fi, Ethernet, Mobile telephony) to build a smart home / smart grid system. This allows devices to talk directly to the cloud-based energy management system as well as other IoT devices. This direct connection permits different manufactures to keep a connection to their devices and their users, and having access to important data of connected devices.

4.2 Architecture overview

4.2.1 Introduction

SPINE-IoT is based upon the OpenAPI specification to model an API that permits interactions with devices. Devices can comprise sub-devices, called entities. The data of the entities is organized in so-called features. A feature is an address with an explicitly named data type.

One of the most important capabilities of SPINE-IoT is the explicit support of Use Cases. Each device can report which Use Cases it supports. Via so-called Use Case interfaces it is possible to work with the device's data (i.e. the features) and hence the functionaility in a more simple way in comparison to work directly with the features: A Use Case interface permits to get specifically data (features) of the requested Use Case and it can provide Use Case specific commands to alter the data (e.g. to trigger a specific behaviour at the device).

The API models all device, entity, and feature information, which permits to analyse the devices in detail. However, a client may also search directly for the supported Use Cases instead. Once a client is configured for selected Use Case instances, the further interactions can basically be focused on the Use Case interfaces.

The rest of this introduction provides some more technical details on the architecture.

The OpenAPI Specification defines a standard, language-agnostic interface to RESTful APIs which allows both humans and computers to discover and understand the capabilities of a service.

An OpenAPI definition can then be used by documentation generation tools to display the API, code generation tools to generate servers and clients in various programming languages, testing tools, and many other use cases [2].

The SPINE-IoT Protocol is an OpenAPI-based specification to model ONE API of a given (cloud) provider with HTTP protocol concepts and JSON payload. It specifies a LIST of devices. Each device can have a LIST of Use Cases. The API model is available as YAML files (see EN 50631-3-1).

Details on device model will be explained in subsequent sections. With regard to a device's functional data, there are in general two interaction styles possible:

- Features: A feature is the core of the API design, it models and represents the application domain. The feature is uniquely defined with by the tuple (deviceId, entityId, featureObjType, featureType).

- Use Case Interfaces: A Use Case Interface retrieves the features of the requested Use Case instances. Many Use Cases require the use of more than one feature.

A Use Case defines actors with particular roles and information (data) exchange to fulfil a defined task. The Use Case discovery allows to discover which Use Cases are supported and which actor a device embodies in a corresponding Use Case. This allows to derive information about which data a device supports as a client or as a server, as defined by each Use Case scenario.

Use Case discovery simplifies finding the right entity address where a server's data of a Use Case can be found. It also permits announcement of Use Cases that are available only conditionally.

Once a client read out present data at the required addresses, it can compare the data with requirements stated by a Use Case specification to see if all requirements are fulfilled for a given scenario.

Next to Use Case discovery and Use case interfaces, SPINE-IoT supports the following mechanisms:

- Discovery: Allows to discover devices and their components and data.

- Binding: A binding is required to obtain particular permissions to control a (part) of a device.

- Subscription: Subscribes to an event stream to monitor for changes.

In this document Oauth2 will be used according to RFC 6749 [3]. Oauth2.0 Authorization Code Grant Flow is used to get an access token. The Oauth2 requires that a client is registered on the authorization server before the client can be granted access to the resources of the resource server. The Oauth2 does not oblige the communication patterns (data and its format) to share necessary information between a resource server and an authorization server. The client needs to use the access token as bearer token in each request to the

API. The security tokens contain information about authorization and are intended to be used for accessing resources on a resource server or refreshing an access token.

Some definitions and values in SPINE-IoT should be identical to SPINE. The identical definitions and values are explained in details in Clause 5.

4.2.2 API versioning

The API SHALL be versioned to allow compatibility between communicating devices. Only as a whole and not in parts, even if several elements of the API are unchanged between different API versions.

SPINE-IoT follows the version numbering scheme according to the “Semantic Versioning 2.0.0” specification [4]. The SPINE-IoT version number has the format MAJOR.MINOR.PATCH, with the major, minor and patch constituents being non-negativ integer numbers as specified in the “Semantic Versioning 2.0.0” specification [4].

This edition defines the major version “1”, with the server variable “majorVersion” set to “v1” accordingly (see YAML models in EN 50631-3-1). This technical solution defines the SPINE-IoT Protocol Specification version 1.0.0.

For a given MAJOR.MINOR.PATCH version, the patch number is increased if and only if the documentation contains backwards compatible corrections to the previous specification version. For a given MAJOR.MINOR version, the minor version is increased if the specification is extended with backwards compatible extensions. For a given MAJOR version, the major version is increased if incompatible changes to the previous version are introduced.

For a given base URL, the server provides exactly one version of the API. This version belongs to exactly one major version. A server may provide other versions of the API at different base URLs.

For information on the API itself and potentially further API versions supported by the server (at other base URLs), this path is available:

□ Path “/api”

Access to information through this path depends on the authorization scope. In particular, a command like “GET /api” needs to return information on other API versions only if the authorized requester is permitted to access the other versions or at least can ask for access to the other versions.

The properties of a successful response are shown in Table 1.

Table 1 — Properties of “api” information

Element name	Type	M/O/C	Brief explanation
thisVersion	string	M	The version number of this API. The format is “MAJOR.MINOR.PATCH” (without quotation marks).
otherVersions	array of string	O	Array of further version numbers supported by the server (at other URLs). For each major version, it is sufficient to specify just the highest version number.

4.3 Device model

4.3.1 General

Like SPINE, a SPINE-IoT device is subdivided into one or more entities (see Clause 5). An entity represents a logical sub-device. Each entity can comprise other entities or so-called “features”. A SPINE-IoT feature is a collection of data, e.g. a measured power value or a table of time-dependent prices, as described in Figure 2.

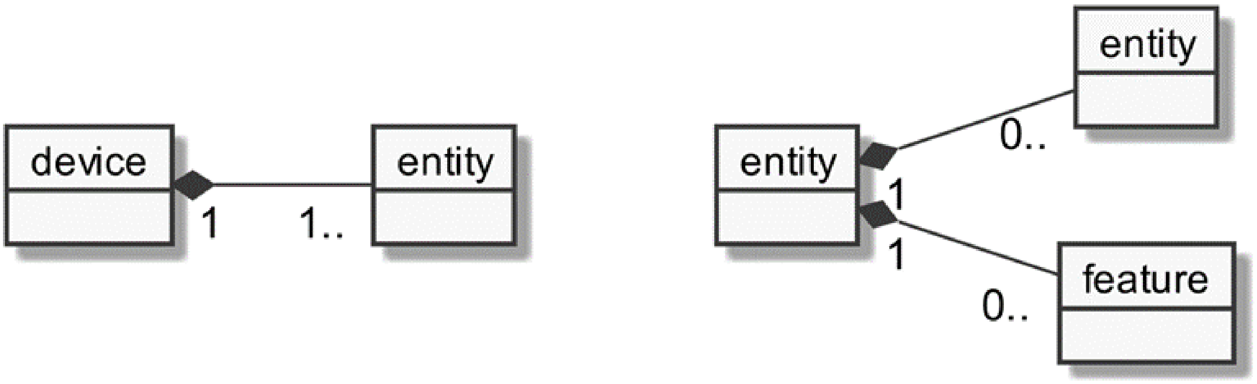

Figure 2 — Primary device model

4.3.2 Device

In the SPINE-IoT model, each level can be accessed individually. For the device information level, this path is available:

— Path “/devices”:

This path provides information on available devices.

The access to information through these paths and other SPINE-IoT paths depends on the authorization scope. In particular, a command like “GET /devices” will only return information on those devices where the authorized requester was granted sufficient access rights.

Within the given API, each device is uniquely identified by its deviceId. The device properties of a successful response are shown in Table 2. Please note that in this subclause only the properties of a single device are described, not the complete response. The full response is described in EN 50631-3-1 (see YAML models).

With a GET command on the above-mentioned paths it is possible to get an overview on available device types and available entity types or to query for a particular device or device type. The deviceId is an essential parameter in queries of almost all paths if information of a particular device is required.

Table 2 — Properties of “device” information

Element name	Type	M/O/C	Brief explanation
deviceId	string	M	SPINE-IoT device identifier. SHALL be unique within the API.
deviceAddress	string	O	SPINE device address. SHALL be unique. See Clause 5.
deviceType	string	M	SPINE device type. See Clause 5.
entitiesOverview.	object	M	Overview of available entity properties.
entitiesOverview. entityTypes	array of string	M	Array of SPINE entity types that are available for the given device. See Clause 5.

4.3.3 Entity

For the entity information level, this path is available:

— Path “/entities”

Access to information through this path depends on the authorization scope. In particular, a command like “GET /entities” will only return information on those entities where the authorized requester was granted