IEC 62541-15

Edition 1.0 2025-02

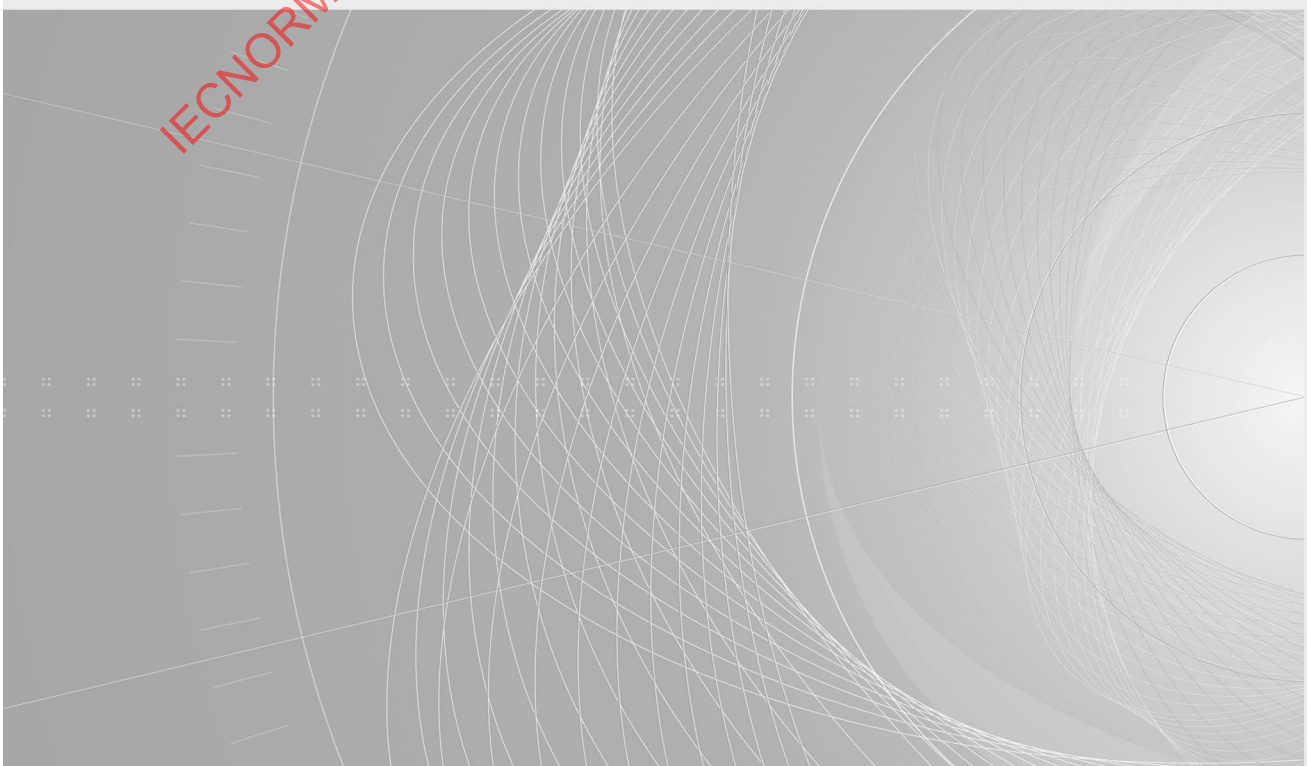
INTERNATIONAL STANDARD

NORME INTERNATIONALE

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

OPC Unified Architecture – Part 15: Safety

Architecture unifiée OPC – Partie 15: Sécurité



IEC 62541-15:2025-02(en-fr)

THIS PUBLICATION IS COPYRIGHT PROTECTED
Copyright © 2025 IEC, Geneva, Switzerland

All rights reserved. Unless otherwise specified, no part of this publication may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in writing from either IEC or IEC's member National Committee in the country of the requester. If you have any questions about IEC copyright or have an enquiry about obtaining additional rights to this publication, please contact the address below or your local IEC member National Committee for further information.

Droits de reproduction réservés. Sauf indication contraire, aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'IEC ou du Comité national de l'IEC du pays du demandeur. Si vous avez des questions sur le copyright de l'IEC ou si vous désirez obtenir des droits supplémentaires sur cette publication, utilisez les coordonnées ci-après ou contactez le Comité national de l'IEC de votre pays de résidence.

IEC Secretariat
3, rue de Varembé
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
info@iec.ch
www.iec.ch

About the IEC
The International Electrotechnical Commission (IEC) is the leading global organization that prepares and publishes International Standards for all electrical, electronic and related technologies.

About IEC publications
The technical content of IEC publications is kept under constant review by the IEC. Please make sure that you have the latest edition, a corrigendum or an amendment might have been published.

IEC publications search - webstore.iec.ch/advsearchform
The advanced search enables to find IEC publications by a variety of criteria (reference number, text, technical committee, …). It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished
Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and once a month by email.

IEC Customer Service Centre - webstore.iec.ch/csc
If you wish to give us your feedback on this publication or need further assistance, please contact the Customer Service Centre: sales@iec.ch.

IEC Products & Services Portal - products.iec.ch
Discover our powerful search engine and read freely all the publications previews, graphical symbols and the glossary. With a subscription you will always have access to up to date content tailored to your needs.

Electropedia - www.electropedia.org
The world's leading online dictionary on electrotechnology, containing more than 22 500 terminological entries in English and French, with equivalent terms in 25 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

A propos de l'IEC
La Commission Electrotechnique Internationale (IEC) est la première organisation mondiale qui élabore et publie des Normes internationales pour tout ce qui a trait à l'électricité, à l'électronique et aux technologies apparentées.

A propos des publications IEC
Le contenu technique des publications IEC est constamment revu. Veuillez vous assurer que vous possédez l'édition la plus récente, un corrigendum ou amendement peut avoir été publié.

Recherche de publications IEC - webstore.iec.ch/advsearchform
La recherche avancée permet de trouver des publications IEC en utilisant différents critères (numéro de référence, texte, comité d'études, …). Elle donne aussi des informations sur les projets et les publications remplacées ou retirées.

IEC Just Published - webstore.iec.ch/justpublished
Restez informé sur les nouvelles publications IEC. Just Published détaille les nouvelles publications parues. Disponible en ligne et une fois par mois par email.

Service Clients - webstore.iec.ch/csc
Si vous désirez nous donner des commentaires sur cette publication ou si vous avez des questions contactez-nous: sales@iec.ch.

IEC Products & Services Portal - products.iec.ch
Découvrez notre puissant moteur de recherche et consultez gratuitement tous les aperçus des publications, symboles graphiques et le glossaire. Avec un abonnement, vous aurez toujours accès à un contenu à jour adapté à vos besoins.

Electropedia - www.electropedia.org
Le premier dictionnaire d'électrotechnologie en ligne au monde, avec plus de 22 500 articles terminologiques en anglais et en français, ainsi que les termes équivalents dans 25 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

IEC 62541-15

Edition 1.0 2025-02

INTERNATIONAL STANDARD

NORME INTERNATIONALE

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

OPC Unified Architecture – Part 15: Safety

Architecture unifiée OPC – Partie 15: Sécurité

INTERNATIONAL ELECTROTECHNICAL COMMISSION

COMMISSION ELECTROTECHNIQUE INTERNATIONALE

ICS 25.040.40

ISBN 978-2-8327-0212-3

Warning! Make sure that you obtained this publication from an authorized distributor.
Attention! Veuillez vous assurer que vous avez obtenu cette publication via un distributeur agréé.

® Registered trademark of the International Electrotechnical Commission
Marque déposée de la Commission Electrotechnique Internationale

CONTENTS

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

INTERNATIONAL ELECTROTECHNICAL COMMISSION

_____

OPC UNIFIED ARCHITECTURE –

Part 15: Safety

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as "IEC Publication(s)"). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
5) IEC itself does not provide any attestation of conformity. Independent certification bodies provide conformity assessment services and, in some areas, access to IEC marks of conformity. IEC is not responsible for any services carried out by independent certification bodies.
6) All users should ensure that they have the latest edition of this publication.
7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
9) IEC draws attention to the possibility that the implementation of this document may involve the use of (a) patent(s). IEC takes no position concerning the evidence, validity or applicability of any claimed patent rights in respect thereof. As of the date of publication of this document, IEC had received notice of (a) patent(s), which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at https://patents.iec.ch. IEC shall not be held responsible for identifying any or all such patent rights.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

IEC 62541-15 has been prepared by subcommittee 65C: Industrial networks, of IEC technical committee 65: Industrial-process measurement, control and automation. It is an International Standard.

The text of this International Standard is based on the following documents:

Draft	Report on voting
65C/1334/FDIS	65C/1339/RVD

Full information on the voting for its approval can be found in the report on voting indicated in the above table.

The language used for the development of this International Standard is English.

This document was drafted in accordance with ISO/IEC Directives, Part 2, and developed in accordance with ISO/IEC Directives, Part 1 and ISO/IEC Directives, IEC Supplement, available at www.iec.ch/members_experts/refdocs. The main document types developed by IEC are described in greater detail at www.iec.ch/publications.

Throughout this document and the referenced other parts of the IEC 62541 series, certain document conventions are used:

Italics are used to denote a defined term or definition that appears in Clause 3 in one of the parts of the series.

Italics are also used to denote the name of a service input or output parameter or the name of a structure or element of a structure that are usually defined in tables.

The *italicized terms* and *names* are also, with a few exceptions, written in camel-case (the practice of writing compound words or phrases in which the elements are joined without spaces, with each element's initial letter capitalized within the compound). For example, the defined term is *AddressSpace* instead of Address Space. This makes it easier to understand that there is a single definition for *AddressSpace*, not separate definitions for Address and Space.

A list of all parts of the IEC 62541 series, published under the general title *OPC Unified Architecture*, can be found on the IEC website.

The committee has decided that the contents of this document will remain unchanged until the stability date indicated on the IEC website under webstore.iec.ch in the data related to the specific document. At this date, the document will be

- reconfirmed,
- withdrawn, or
- revised.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

INTRODUCTION

OPC UA safety extends OPC UA to fulfill the requirements of functional safety as defined in the IEC 61508 series and IEC 61784-3 series of standards.

Figure 1 shows the relationship between this document and the relevant safety and OPC UA standards in an industrial environment. An arrow from Document A to Document B means "Document A is referenced in Document B". This reference can be either normative or informative. Not all of these standards are applicable or required for a given product.

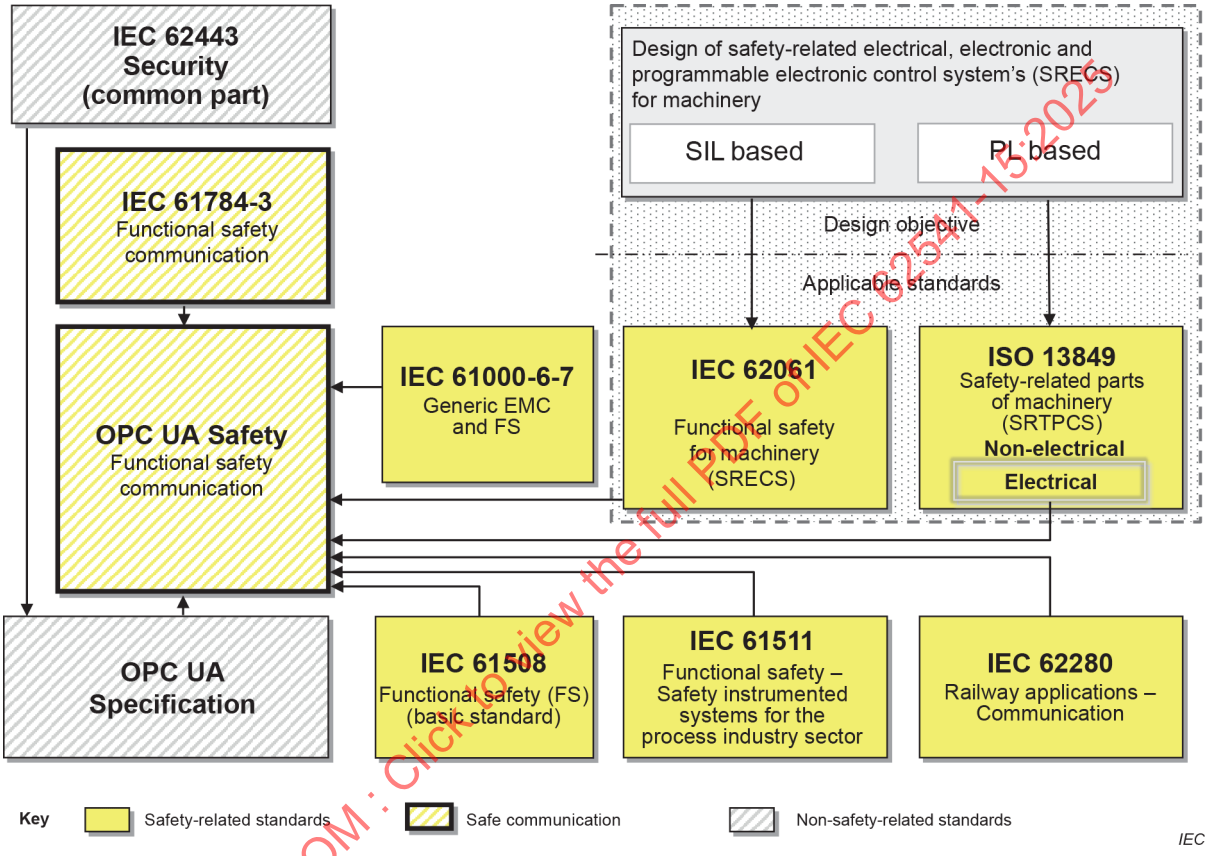


Figure 1 – Relationships of OPC UA safety with other standards

Implementing this document allows for detecting all types of communication errors encountered in the lower network layers. In case an error is detected, this information is shared with the safety applications in the user layer which can then act in an appropriate way, e.g. by switching to a safe state.

The document describes the behaviour of the individual endpoints for safe communication, as well as the OPC UA *Information Model* which is used to access these endpoints.

This document is application-independent and does not pose requirements on the structure and length of the application data. Application-specific requirements are expected to be described in appropriate companion specifications.

This document can be used for applications requiring functional safety up to the *safety integrity level* (*SIL*) 4.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

OPC UNIFIED ARCHITECTURE –

Part 15: Safety

1 Scope

This document describes a *safety communication layer* (services and a protocol) for the exchange of *SafetyData* using IEC 62541 mechanisms. It identifies the principles for functional safety communications defined in IEC 61784-3 that are relevant for this *safety communication layer*. This *safety communication layer* is intended for implementation in *safety* devices only.

NOTE 1 This document targets controller-to-controller communication. However, easy expandability to other use-cases (e.g. OPC UA field level communication) has already been considered in the design of this document.

NOTE 2 This document does not cover electrical safety and intrinsic safety aspects. Electrical safety relates to hazards such as electrical shock. Intrinsic safety relates to hazards associated with potentially explosive atmospheres.

This document defines mechanisms for the transmission of safety-relevant messages among participants within a network using OPC UA technology in accordance with the requirements of the IEC 61508 series and IEC 61784-3 for functional safety. These mechanisms can be used in various industrial applications such as process control, manufacturing, automation, and machinery.

This document provides guidelines for both developers and assessors of compliant devices and systems.

NOTE 3 The resulting *SIL* claim of a system depends on the implementation of this document within the system – implementation of this document in a standard device is not sufficient to qualify it as a safety device.

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.

IEC 61508 (all parts), *Functional safety of electrical/electronic/programmable electronic safety-related systems*

IEC 61784-3:2021, *Industrial communication networks – Profiles – Part 3: Functional safety fieldbuses – General rules and profile definitions*

IEC 62443 (all parts), *Industrial communication networks – Network and system security*

IEC 62541-1:2020, *OPC Unified Architecture – Part 1: Overview and Concepts*

IEC 62541-3:2020, *OPC Unified Architecture – Part 3: Address Space Model*

IEC 62541-4:2020, *OPC Unified Architecture – Part 4: Services*

IEC 62541-5:2020, *OPC Unified Architecture – Part 5: Information Model*

IEC 62541-6:2020, *OPC Unified Architecture – Part 6: Mappings*

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

IEC 62541-14, *OPC Unified Architecture – Part 14: PubSub*

ISO/IEC 9834-8:2014, *Information technology – Procedures for the operation of object identifier registration authorities – Part 8: Generation of universally unique identifiers (UUIDs) and their use in object identifiers*

3 Terms, definitions, symbols, abbreviated terms and conventions

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC 62541-1:2020, IEC 62541-3:2020, IEC 62541-4:2020, IEC 62541-6:2020 and the following apply.

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

- IEC Electropedia: available at https://www.electropedia.org/
- ISO Online browsing platform: available at https://www.iso.org/obp

NOTE This document uses concepts of IEC 62541 information modeling to describe the concepts in this document.

3.1.1 Common terms and definitions

3.1.1.1
Cyclic Redundancy Check
CRC
<value> redundant data derived from, and stored or transmitted together with, a block of data in order to detect data corruption

<method> procedure used to calculate the redundant data

Note 1 to entry: Terms "CRC code" and "CRC signature", and labels such as CRC1, CRC2, may also be used in this document to refer to the redundant data.

[SOURCE: IEC 61784-3:2021, 3.10]

3.1.1.2
error
discrepancy between a computed, observed or measured value or condition and the true, specified or theoretically correct value or condition

Note 1 to entry: Errors may be due to design mistakes within hardware/software and/or corrupted information due to electromagnetic interference and/or other effects.

Note 2 to entry: Errors do not necessarily result in a failure or a fault.

[SOURCE: IEC 60050-192:2024, 192-03-02, modified – notes added]

3.1.1.3
failure
termination of the ability of a functional unit to perform a required function or operation of a functional unit in any way other than as required

Note 1 to entry: Failure can be due to an error (for example, problem with hardware/software design or message disruption).

[SOURCE: IEC 61508-4:2010, 3.6.4, modified – notes and figures deleted, new note to entry added]

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

3.1.1.4
fault
abnormal condition that may cause a reduction in, or loss of, the capability of a functional unit to perform a required function

Note 1 to entry: IEV 191-05-01 defines "fault" as a state characterized by the inability to perform a required function, excluding the inability during preventive maintenance or other planned actions, or due to lack of external resources.

[SOURCE: IEC 61508-4:2010, 3.6.1, modified – figure reference deleted]

message
<information theory and communication theory>ordered sequence of characters (usually octets) intended to convey information

[SOURCE: ISO/IEC 2382:2015, 2123031, modified – insertion of "(usually octets)", deletion of notes and source]

3.1.1.5
performance level
PL
discrete level used to specify the ability of safety-related parts of control systems to perform a safety function under foreseeable conditions

[SOURCE: ISO 13849-1:2023, 3.1.5]

3.1.1.6
residual error probability
probability of an error undetected by the *SCL* safety measures

[SOURCE: IEC 61784-3:2021, 3.1.35]

3.1.1.7
residual error rate
statistical rate at which the *SCL* safety measures fail to detect errors

[SOURCE: IEC 61784-3:2021, 3.1.36]

3.1.1.8
safety communication layer
SCL
communication layer above the IEC 62541 communication stack that includes all necessary additional measures to ensure safe transmission of data in accordance with the requirements of IEC 61508

Note 1 to entry: The *SCL* provides several services, the most important ones being the *SafetyProvider* and the *SafetyConsumer*.

[SOURCE: IEC 61784-3:2021, 3.1.39, modified – "FAL" replaced by "IEC 62541 communication stack", note to entry added]

3.1.1.9
safety function response time
worst case elapsed time following an actuation of a safety sensor connected to a fieldbus, until the corresponding safe state of its safety actuator(s) is achieved in the presence of errors or failures in the safety function

Note 1 to entry: This concept is introduced in IEC 61784-3:2021, 5.2.4 and is addressed by the functional safety communication profiles defined in the IEC 61784-3 series of documents.

[SOURCE: IEC 61784-3:2021, 3.1.44]

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

3.1.1.10
safety integrity level
SIL
discrete level (one out of a possible four), corresponding to a range of safety integrity values, where safety integrity level 4 has the highest level of safety integrity and safety integrity level 1 has the lowest

Note 1 to entry: The target failure measures (see IEC 61508-4:2010, 3.5.17) for the four *safety integrity levels* are specified in Table 2 and Table 3 of IEC 61508-1:2010.

Note 2 to entry: *Safety integrity levels* are used for specifying the safety integrity requirements of the safety functions to be allocated to the E/E/PE safety-related systems.

Note 3 to entry: A *safety integrity level* (*SIL*) is not a property of a system, subsystem, element or component. The correct interpretation of the phrase "*SIL* n safety-related system" (where n is 1, 2, 3 or 4) is that the system is potentially capable of supporting safety functions with a *safety integrity level* up to n.

[SOURCE: IEC 61508-4:2010, 3.5.8]

3.1.1.11
safety measure
measure to control possible communication errors that is designed and implemented in compliance with the requirements of IEC 61508

Note 1 to entry: In practice, several safety measures are combined to achieve the required *safety integrity level*.

Note 2 to entry: Communication errors and related safety measures are detailed in IEC 61784-3:2021, 5.3 and 5.4.

[SOURCE: IEC 61784-3:2021, 3.1.46]

3.1.1.12
safety PDU
SPDU
PDU transferred through the *safety communication channel*

Note 1 to entry: The SPDU may include more than one copy of the *SafetyData* using differing coding structures and hash functions together with explicit parts of additional protections such as a key, a sequence count, or a time stamp mechanism.

Note 2 to entry: Redundant *SCLs* may provide two different versions of the SPDU for insertion into separate fields of the IEC 62541 frame.

[SOURCE: IEC 61784-3:2021, 3.1.47]

3.1.2 Additional terms and definitions

3.1.2.1
fail-safe
ability of a system that, by adequate technical or organizational measures, prevents from hazards either deterministically or by reducing the risk to a tolerable measure

Note 1 to entry: Equivalent to functional safety.

3.1.2.2
fail-safe substitute values
FSV
values which are issued or delivered instead of process values when the safety function is set to a fail-safe state

Note 1 to entry: In this document, the fail-safe substitute values (FSV) are always set to binary "0".

3.1.2.3
flag
one-bit value used to indicate a certain status or control information

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

3.1.2.4
globally unique identifier
GUID
128-bit number used to identify information in computer systems

Note 1 to entry: The term universally unique identifier (UUID) is also used.

Note 2 to entry: In this document, UUID version 4 is used.

3.1.2.5
MonitoringNumber
MNR
means used to ensure the correct order among transmitted safety PDUs and to monitor the communication delay

Note 1 to entry: Instance of sequence number as described in IEC 61784-3.

Note 2 to entry: The *MNR* starts at a random value and is incremented with each request. It rolls over to a minimum threshold value that is not zero.

Note 3 to entry: The transmitted *MNR* is protected by the transmitted *CRC* signature of the ResponseSPDU.

3.1.2.6
non-safety-
predicate meaning that the respective object is a "standard" object and has not been designed and implemented to fulfil any requirements with respect to functional safety

3.1.2.7
OPC UA mapper
non-safety-related part of the implementation of this document which maps the SPDU to the actual IEC 62541 services

Note 1 to entry: Depending on which services of IEC 62541 are being used (e.g. *Client/Server* or *PubSub*), different mappers can be specified.

3.1.2.8
process values
PV
input and output data (in a safety PDU) that are required to control an automated process

3.1.2.9
qualifier
attribute (bit or Boolean), indicating whether the corresponding value is valid or not (e.g. being a fail-safe substitute value)

3.1.2.10
SafetyAutomationComponent
SafetyAC
communication partner in a unidirectional safety link

Note 1 to entry: A *SafetyAutomationComponent* can be a *SafetyProvider* (data source), a *SafetyConsumer* (data sink), or both.

3.1.2.11
SafetyConsumer
entity (usually software) that implements the data sink of a unidirectional safety link

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

3.1.2.12
SafetyData
application data transmitted across a safety network using a safety protocol

Note 1 to entry: The *safety communication layer* does not ensure the safety of the data itself, but only that the data is transmitted safely.

3.1.2.13
SafetyProvider
entity (usually software) that implements the data source of a unidirectional safety link

3.1.2.14
SafetyBaseID
randomly generated authenticity ID which is used to safely authenticate *SafetyProviders* having the same *SafetyProviderID*

Note 1 to entry: Together with the *SafetyProviderID*, it is an instance of connection authentication as described in IEC 61784-3.

3.1.2.15
SafetyProviderID
user-assigned, locally unique identifier which is used to safely authenticate *SafetyProviders* within a certain area

Note 1 to entry: Together with the *SafetyBaseID*, it is an instance of connection authentication as described in IEC 61784-3.

Note 2 to entry: All *SafetyProviders* within an area such defined may share an identical *SafetyBaseID*.

3.1.2.16
standard transmission system
part of the transmission system (implemented in hardware and software) that is not implemented according to any safety standards

Note 1 to entry: This document is using the services of the *standard transmission system* to transmit prebuilt safety packets.

3.2 Symbols and abbreviated terms

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

3.2.1 Abbreviated terms from IEC 61784-3

CRC	cyclic redundancy check	
PDU	protocol data unit	[ISO/IEC 7498-1]
PL	performance level	[ISO 13849-1]
PLC	programmable logic controller	
SCL	safety communication layer	
SIL	safety integrity level	[IEC 61508-4]
SPDU	safety PDU, safety protocol data unit	

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

3.2.2 Additional symbols and abbreviated terms

3.2.2.1 Abbreviated terms

FSV	fail-safe substitute values
HMI	human-machine interface
ID	identifier
LSB	least significant bit
MNR	MonitoringNumber
MSB	most significant bit
OA	operator acknowledgment
OPC UA PI	OPC UA platform interface
PI	platform interface
PV	process values
SAPI	safety application program interface
SFRT	safety function response time
SPI	safety parameter interface
STrailer	safety trailer
TRA	threat and risk analysis

3.2.2.2 Symbols

p	bit error probability
$P_{re,cond}$	conditional residual error probability

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

3.3 Conventions

3.3.1 General conventions

Terms or names where two capital letters of abbreviations are in sequence or for separation to a suffix are written with underscores in between.

The abbreviation "F" is an indication for *safety-related* items, technologies, systems, and units (*fail-safe*, functional safe).

The default data that are used in case of unit failures or errors, are called *fail-safe substitute values* (*FSV*) and are set to binary "0".

Reserved bits ("res") are set to "0" and ignored by the receiver to avoid problems with future versions of this document.

The notation 0x… represents a hexadecimal value.

3.3.2 Conventions for requirements numbering

Requirements in this document are designated as *[RQx.yz]*, where *x* denotes the chapter number, *y* is a counter and *z* is an optional character to link closely related requirements. The following are examples of valid requirements designations: [RQ8.15] (requirement 15 in chapter 8); [RQ47.11a], [RQ47.11b] (requirements 11a and 11b in chapter 47, which are closely related).

The initial numbering of requirements was chosen such that counters within each chapter are in ascending order. However, the addition of further requirements leads to deviations from this rule since existing requirements shall keep their initial designation.

For an informative index of all the requirements in this document, see 10.3.

3.3.3 Conventions in state machines

See Table 1 for the conventions used in state machines.

Table 1 – Conventions used in state machines

Convention	Meaning
:=	Assignment: value of an item on the left is replaced by value of the item on the right.
<	Less than: a logical condition yielding TRUE if and only if an item on the left is less than the item on the right.
<=	Less or equal than: a logical condition yielding TRUE if and only if an item on the left is less or equal than the item on the right.
>	Greater than: a logical condition yielding TRUE if and only if the item on the left is greater than the item on the right.
>=	Greater or equal than: a logical condition yielding TRUE if and only if the item on the left is greater or equal than the item on the right.
==	Equality: a logical condition yielding TRUE if and only if the item on the left is equal to an item on the right.
<>	Inequality: a logical condition yielding TRUE if and only if the item on the left is not equal to an item on the right.
&&	Logical "AND" (Operation on binary values or results).
||	Logical "OR" (Operation on binary values or results).
⊕	Logical "XOR" (Operation on binary values or digital values).
[..]	UML Guard condition, if and only if the guard is TRUE the respective transition is enabled.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

4 Overview of OPC UA Safety

4.1 General

This document specifies a *safety communication layer* (*SCL*) allowing safety-related devices to use the services of IEC 62541 for the safe exchange of safety-related data. A safety device that implements OPC UA Safety correctly will be able to exchange safety-related data and hereby fulfill the requirements of the IEC 61508 series and IEC 61784-3. This document uses a *MonitoringNumber*, a timeout, a set of IDs and a *cyclic redundancy check* (*CRC*) for the detection of all possible communication errors which can happen in the underlying IEC 62541 *standard transmission system.* These *safety measures* have been quantitatively evaluated and offer a probability of dangerous failure per hour (PFH) and a probability of dangerous failure on demand (PFD) sufficing to build *safety-related* applications with a *safety integrity level* of up to *SIL*4.

OPC UA Safety itself is an application-independent, general solution. The length and structure of the data sent is defined by the safety application. However, application-dependent companion specifications (addressing for example electro-sensitive protective equipment, electric drives with safety functions, forming presses, robot safety, and automated guided vehicles) are expected to be defined by application-experts in appropriate OPC UA companion specifications.

4.2 Implementation aspects

[RQ4.1] All technical measures for error detection described in this document shall be implemented within the *SCL* in devices designed in accordance with the IEC 61508 series and shall meet the target *SIL*.

4.3 Features

- Runs on top of:
  - IEC 62541 *Client/Server* with the *Method Service Set*.
  - IEC 62541 *PubSub*.
- From an architectural point of view: easy extensibility for other ways of communication.
- Goal: no modification of existing OPC UA framework.
- The state machines of this document are independent from the *OPC UA mapper*, allowing for a simplified exchange of the mapper.
- Ready for wireless transmission channels.
- Modest requirements on safety network nodes:
  - No clock synchronization is necessary (no requirements regarding the accuracy between clocks at different nodes).
  - Within the *SafetyConsumer*, a safety-related, local timer is required for implementing the *SafetyConsumerTimeout*. The accuracy of this timer depends on the timing requirements of the safety application.
- End-to-End Safety: functional *SafetyData* is transported between two safety endpoint devices across a standard network that is not functionally safety compliant. This includes the lower transport layers such as the IEC 62541 stack, underlying physical media, and *non-safety* network elements (e.g. routers and switches).
- "Dynamic" systems:
  - Safety communication partners can change during runtime,
  - either an increase or decrease, or both, in the number of safety communication partners can occur.
- Well-defined text-strings are used for diagnostic purposes.
- Safety communication and standard communication are independent. However, standard devices and safety devices can use the same *standard transmission system* at the same time.
- Functional safety can be achieved without using structurally redundant *standard transmission system*s i.e. a single channel approach can be used. Redundancy can be used optionally for increased availability.
- For diagnostic purposes, the last *SPDU* sent and received is accessible in the *Information Model* of the *SafetyProvider*.
- Length of user data: 1 octet to 1 500 octets, structures of basic *DataTypes*, see 6.2.5.

4.4 Security policy

In the final application, an appropriate security environment is necessary to protect both the operational environment and the safety-related systems.

This document does not cover security aspects, nor does it provide any requirements for security.

A threat and risk analysis (TRA) according to the IEC 62443 series shall be carried out on a final application system level.

During compliance tests for this document, security aspects are not part of the scope, as it is assumed that the underlying base mechanisms (i.e. *Methods*) already provide adequate security.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

5 General

5.1 External documents providing specifications for the profile

No other external documents are providing specifications in addition to the Normative references given in Clause 2.

5.2 Safety functional requirements

The following requirements apply for the development of this document:

a) Safety communication suitable for *safety integrity level* up to *SIL*4 (see the IEC 61508 series) and PL e (see ISO 13849-1).

b) Combination of *SIL* 1 to 4 devices according to this document as well as *non-safety* devices on one communication network.

c) Implementation of the safety transmission protocol is restricted to the safety layer.

d) The safety-relevant time-out monitoring is implemented in the safety layer.

e) Safety communication meet the requirements of IEC 61784-3.

f) [RQ5.1] This document is intended for implementation in safety devices exclusively. Exceptions (e.g. for debugging, simulation, testing, and commissioning) shall be discussed with a notified body.

5.3 Safety measures

[RQ5.2] For an implementation of this document, the following *safety measures* shall be implemented: *MonitoringNumber*; timeout with receipt in the *SafetyConsumer*; set of IDs for the *SafetyProvider*; Data Integrity check.

Together, these *safety measures* address all possible transmission errors as listed in IEC 61784-3:2021, 5.5, see Table 2.

[RQ5.3] The *safety measures* shall be processed and monitored within the *SCL*.

Table 2 – Deployed safety measures to detect communication errors

Communication error	Safety measures			
	MonitoringNumber [a]	Timeout with receipt [b]	Set of IDs for SafetyProvider [c]	Data integrity check [d]
Corruption	–	–	–	X
Unintended repetition	X	X	–	–
Incorrect sequence	X	–	–	–
Loss	X	X	–	–
Unacceptable delay	–	X	–	–
Insertion	X	–	–	–
Masquerade	X	–	X	X
Addressing	–	–	X	–

[a] Instance of "sequence number" of IEC 61784-3.

[b] Instance of "time expectation" (timeout) and "feedback message" (receipt) of IEC 61784-3.

[c] Instance of "connection authentication" of IEC 61784-3.

[d] Instance of "data integrity assurance" of IEC 61784-3, based on *CRC* signature.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

The *SafetyConsumer* is specified in such a way that for any communication error according to Table 2, a defined fault reaction will occur.

In all cases, the faulty *SPDU* will be discarded, and not forwarded to the safety application.

Moreover, if the error rate is too high, the *SafetyConsumer* is defined in such a way that it will cease to deliver actual *process values* to the safety application but will deliver *fail-safe substitute values* instead. In addition, an indication at the *Safety Application Program Interface* is set which can be queried by the safety application.

In case the error rate is still considered acceptable, the state machine repeats the request, see 9.4.

5.4 Safety communication layer structure

This document is based on:

- the *standard transmission system* according to IEC 62541
- an additional safety transmission protocol on top of this *standard transmission system*

Safety applications and standard applications share the same standard IEC 62541 communication systems at the same time. The safe transmission function incorporates *safety measures* to detect faults or hazards that originate in the *standard transmission system* which have a potential to compromise the safety subsystems. This includes faults such as:

- Random errors, for example due to electromagnetic interference on the transmission channel;
- Failures or faults of the standard hardware;
- Systematic malfunctions of components within the standard hardware and software.

This principle delimits the assessment effort to the "safe transmission functions". The *standard transmission system* does not require any additional functional safety assessment.

The basic communication layers of this document are shown in Figure 2.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

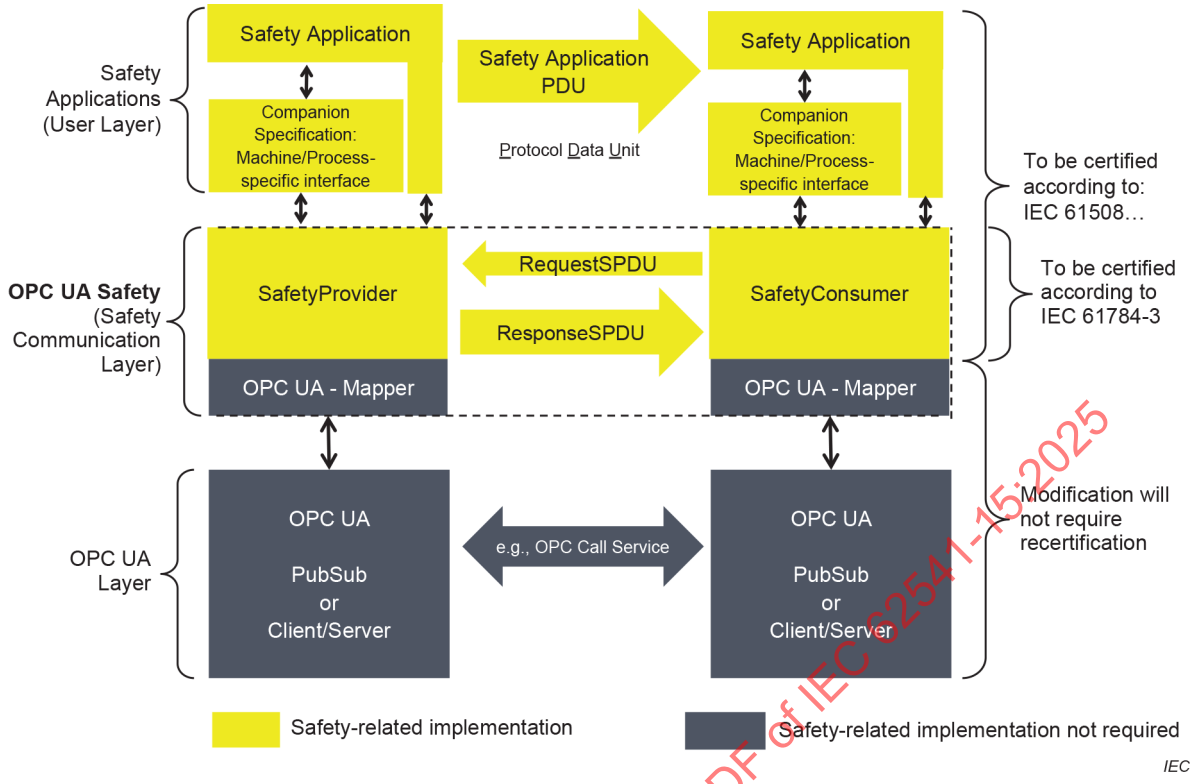


Figure 2 – Safety layer architecture

Summary of the architecture:

Part: User layer

The safety applications in the User Layer are either directly connected to the *SafetyProvider* or *SafetyConsumer*, or they are connected via a machine-specific or process-specific interface, which is described in companion specifications (e.g. sectoral).

The safety applications are expected to be designed and implemented according to the IEC 61508 series.

The Safety applications in the User Layer are not within the scope of this document.

Part: OPC UA Safety (Safety Communication Layer)

This layer is within the scope of this document. It defines the two services *SafetyProvider* and *SafetyConsumer* as basic building blocks. Together, they form the *safety communication layer* (*SCL*), implemented in a safety-related way according to the IEC 61508 series.

SafetyData is transmitted using point-to-point communication (unidirectional). Each unidirectional data flow internally communicates in both directions, using a request and response pattern. This allows for checking the timeliness of messages using a single clock in the *SafetyConsumer*, thus eliminating the necessity for synchronized clocks.

When *SafetyConsumers* connect to *SafetyProviders*, they have prior expectations regarding the pair of *SafetyProviderID* and *SafetyBaseID* (e.g. by configuration). If this expectation is not fulfilled by the *SafetyProvider*, *fail-safe substitute values* are delivered to the safety application instead of the received *process values*. In contrast, it is not necessary for a *SafetyProvider* to know the *SafetyConsumerID* of the *SafetyConsumer* and will provide its *process values* to any *SafetyConsumer* requesting it.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

SafetyProviders can not detect communication errors. All required error detection is performed by the *SafetyConsumer*.

If it is necessary for a pair of safety applications to exchange SafetyData in both directions, two pairs of *SafetyProviders* and *SafetyConsumers* shall be established, one pair for each direction.

Refer to B.2 in Annex B for use cases how *SafetyProviders* and *SafetyConsumers* can be connected.

The OPC UA mapper implements the parts of the *safety communication layer* which are specific for the IEC 62541 communication *Service* in use, i.e. *PubSub* or *Client/Server*. Therefore, the remaining parts of the *safety communication layer* can be implemented independent of the IEC 62541 *Service* being used.

Part: OPC UA Layer

Client/Server:

- The *SafetyProvider* is implemented using an IEC 62541 *Server* providing a *Method*.
- The *SafetyConsumer* is implemented using an IEC 62541 *Client* calling the *Method* provided by the *SafetyProvider*.

PubSub:

- The *SafetyProvider* publishes the *ResponseSPDU* and subscribes to the *RequestSPDU*.
- The *SafetyConsumer* publishes the *RequestSPDU* and subscribes to the *ResponseSPDU*.

5.5 Requirements for CRC calculation

[RQ5.4] Any *CRC* signature calculation shall start with a preset value of "1".

[RQ5.5] Any *CRC* signature calculation resulting in a "0" value, shall use the value "1" instead.

[RQ5.6] *SPDUs* with all values (incl. *CRC* signature) being zero shall be ignored by the receiver (*SafetyConsumer* and *SafetyProvider*).

6 Safety communication layer services

6.1 General

Clause 6 describes the integration of this document into the IEC 62541 *Information Models* in 6.2 and the resulting *Service* interfaces in 6.3. Diagnostic services are described in 6.4.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

6.2 Information models

6.2.1 General

Subclause 6.2 describes the identifiers, types and structure of the *Objects* and *Methods* that are used to implement the *OPC UA mappers* defined in this document. This implementation serves three purposes:

- support of the safe exchange of *SPDUs* at runtime
- online browsing, to identify *SafetyConsumers* and *SafetyProviders*, and to check their parameters for diagnostic purposes
- offline engineering: the *Information Model* of one controller can be exported in a standardized file on its engineering system, be imported in another engineering system, and finally deployed on another controller. This allows for a vendor-independent exchange of the communication interfaces of safety applications, e.g. for establishing connections between devices.

NOTE Neither online browsing nor offline engineering currently supports any features to detect errors. Hence, no guarantees with respect to functional safety are made. This means that online browsing can only be used for diagnostic purposes, and not for exchanging safety-relevant data. It is assumed that errors can occur during the transfer of the *Information Model* from one engineering system to another in the context of offline engineering. Therefore, the programmer of the safety application is responsible for the verification and validation of the safety application.

Consequently, all type values described in 6.2 are defined as read-only, i.e. they cannot be written by general IEC 62541 write commands.

6.2.2 Object and ObjectType Definitions

6.2.2.1 SafetyACSet Object

[RQ6.1] Each *Server* shall have a singleton *Folder* called *SafetyACSet* with a fixed *NodeID* in the *Namespace* of this document. Because all *SafetyProviders* and *SafetyConsumers* on this *Server* contain a hierarchical *Reference* from this *Object* to themselves, it can be used to directly access all *SafetyProviders* and *SafetyConsumers*. *SafetyACSet* is intended for safety-related purposes only. It should not reference *non-safety*-related items.

See Table 3 for the definition of the *SafetyACSet*.

Table 3 – SafetyACSet definition

Attribute	Value		
BrowseName	SafetyACSet		
References	**NodeClass**	**BrowseName**	**Comment**
OrganizedBy by the Objects Folder defined in IEC 62541-5.			
HasTypeDefinition	ObjectType	FolderType	Entry point for all *SafetyProviders* and *SafetyConsumers*
Conformance units			
SafetyACSet			

[RQ6.2] In addition, a *Server* shall comprise one IEC 62541 *Object* derived from *DataType SafetyProviderType* for each *SafetyProvider* it implements, and one IEC 62541 *Object* derived from *DataType SafetyConsumerType* for each *SafetyConsumer* it implements. The corresponding *Information Models* shown in Figure 3 and Figure 4 shall be used.

A description of the graphical notation for the different types of Nodes and references (shown in Figure 3, Figure 4, and Figure 6) can be found in IEC 62541-3.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Figure 3 describes the *SafetyProvider* and the *SafetyConsumer*.

NOTE 1 This document assumes (atomic) consistent data exchange between OPC mappers of the two endpoints.

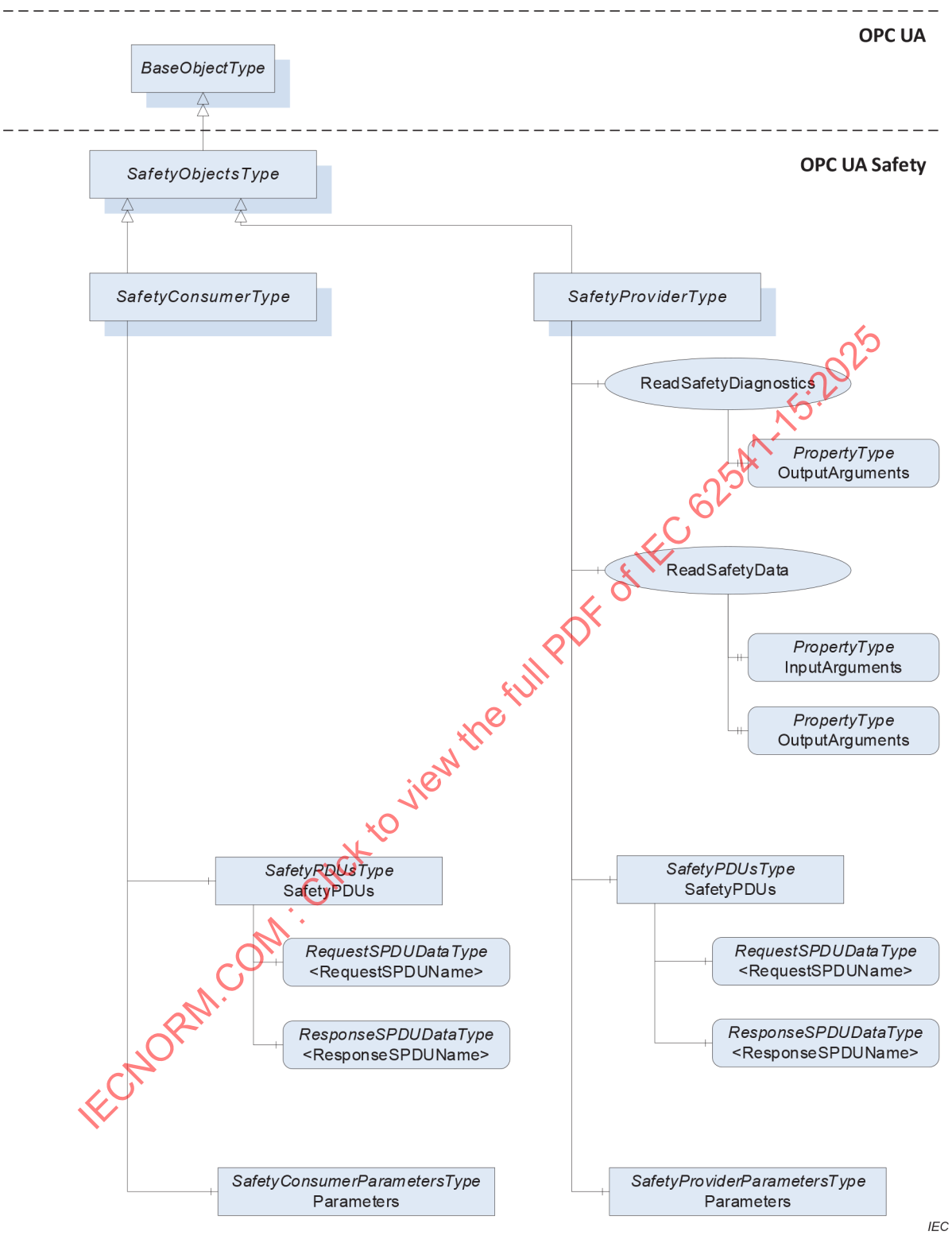
[RQ6.3a] For implementations supporting IEC 62541 *Client/Server*, the *Call Service* of the *Method Service Set* (see IEC 62541-4) shall be used. The *Method ReadSafetyData* has a set of input arguments that make up the *RequestSPDU* and a set of output arguments that make up the *ResponseSPDU*. The *SafetyConsumer* uses the IEC 62541 Client with the IEC 62541 *Service Call*.

[RQ6.3b] For implementations supporting IEC 62541 *PubSub*, the IEC 62541 *Object SafetyPDUs* with its *Properties RequestSPDU* and *ResponseSPDU* shall be used. *RequestSPDU* is published by the *SafetyConsumer* and subscribed by the *SafetyProvider*. *ResponseSPDU* is published by the *SafetyProvider* and subscribed by the *SafetyConsumer*.

NOTE 2 The terms "request" and "response" refer to the behaviour on the layer of this document. Within the *PubSub* context, both requests and responses are realized by repeatedly publishing and subscribing *Messages*, see Figure 14.

[RQ6.4] For diagnostic purposes, the *SPDUs* received and sent shall be accessible by calling the *Method ReadSafetyDiagnostics*.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

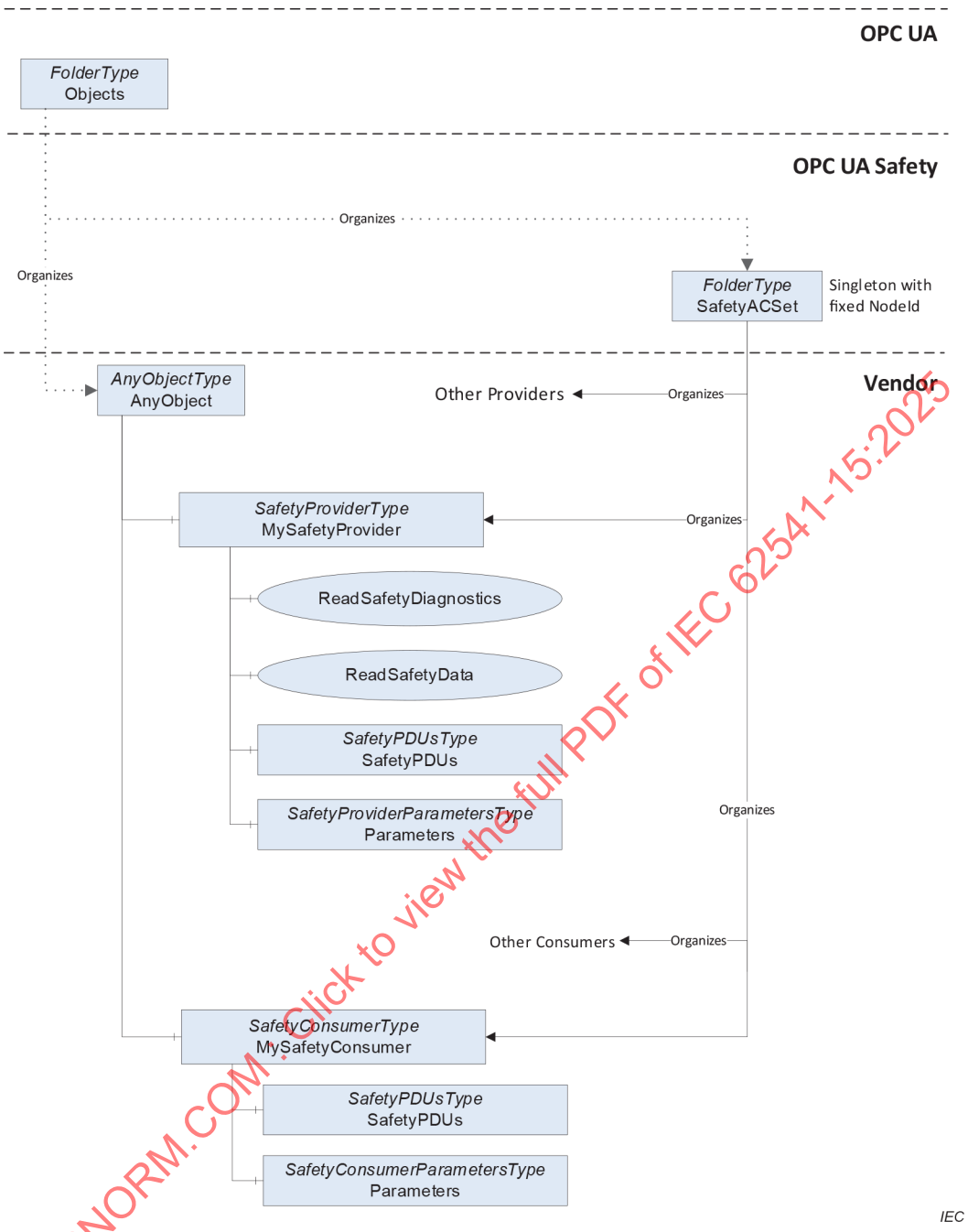


IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Figure 3 – Server Objects for OPC UA Safety

NOTE 3 For the input/output arguments of the *Methods ReadSafetyData* and *ReadSafetyDiagnostics*, see 6.2.2.3 and 6.2.2.4. For the parameters of the *SafetyProvider* and *SafetyConsumer*, see Figure 6, Table 12, and Table 13. For *RequestSPDU* and *ResponseSPDU*, see Table 7, Table 18, Table 20, and 7.2.1.

Figure 4 shows the instances of *Server Objects* for this document. The *ObjectType* for the *SafetyProviderType* contains *Methods* having outputs of the abstract *DataType Structure*. Each instance of a *SafetyProvider* requires its own copy of the *Methods* which contain the concrete *DataTypes* for *OutSafetyData* and *OutNonSafetyData*.



IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Figure 4 – Instances of Server Objects for this document

6.2.2.2 Safety ObjectType definitions

[RQ6.5] To reduce the number of variations and to alleviate validation testing, the following restrictions apply to instances of *SafetyProviderType* and *SafetyConsumerType* (or instances of *DataTypes* derived from *SafetyProviderType* or *SafetyConsumerType*):

1) The references shown in Figure 4 originating at *SafetyProviderType* or *SafetyConsumerType* and below shall be of *ReferenceType HasComponent* (and shall not be derived from *ReferenceType HasComponent*) for *Object References* or *ReferenceType HasProperty* (and shall not be derived from *ReferenceType HasProperty*) for *Property References*.
2) As *BrowseNames* (i.e. name and *Namespace*) are used to find *Methods*, the names of *Objects* and *Properties* shall be locally unique.

3) The *DataType* of both *Properties* and *MethodArguments* shall be used as specified, and no derived *DataTypes* shall be used (exception: *OutSafetyData* and *OutNonSafetyData*).

4) In IEC 62541, the order of *Method* arguments is relevant.

See Table 4 for the definition of the *SafetyObjectsType*.

Table 4 – SafetyObjectsType definition

Attribute	Value				
BrowseName	SafetyObjectsType				
IsAbstract	True				
References	**Node class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling rule**
Subtype of BaseObjectType					
Conformance units					
SafetySupport					

See Table 5 for the definition of the *SafetyProviderType*.

Table 5 – SafetyProviderType definition

Attribute	Value				
BrowseName	SafetyProviderType				
IsAbstract	False				
References	**Node class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling rule**
Subtype of SafetyObjectsType					
HasComponent	Method	ReadSafetyData			Optional
HasComponent	Method	ReadSafetyDiagnostics			Optional
HasComponent	Object	SafetyPDUs		SafetyPDUsType	Optional
HasComponent	Object	Parameters		SafetyProviderParametersType	Mandatory
Conformance units					
SafetyProviderParameters					

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

[RQ6.6] Instances of *SafetyProviderType* shall use non-abstract *DataTypes* for the arguments *OutSafetyData* and *OutNonSafetyData*.

See Table 6 for the definition of the *SafetyConsumerType*.

Table 6 – SafetyConsumerType definition

Attribute	Value				
BrowseName	SafetyConsumerType				
IsAbstract	False				
References	**Node class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling rule**
Subtype of SafetyObjectsType					
HasComponent	Object	SafetyPDUs		SafetyPDUsType	Optional
HasComponent	Object	Parameters		SafetyConsumerParameters Type	Mandatory
Conformance units					
SafetyConsumerParameters					

6.2.2.3 Method ReadSafetyData

This *Method* is mandatory for the *Facet SafetyProviderServerMapper*. It is used to read *SafetyData* from the *SafetyProvider*. It is in the responsibility of the safety application that this *Method* is not concurrently called by multiple *SafetyConsumers*. Otherwise, the *SafetyConsumer* can receive invalid responses resulting in a safe reaction which can lead to either spurious trips or system unavailability, or both.

See Table 7 for *Method ReadSafetyData*'s arguments and Table 8 for its *AdressSpace* definition.

The *Method* argument *OutSafetyData* has an application-specific *DataType* derived from *Structure*. This *DateType* (including the *DataTypeID*) is expected to be the same in both the *SafetyProvider* and the *SafetyConsumer*. Otherwise, the *SafetyConsumer* will not accept the transferred data and switch to *fail-safe substitute values* instead (see state S16 in Table 34 as well as 7.2.3.2 and 7.2.3.5). The *Method* argument *OutNonSafetyData* has an application-specific *DataType* derived from *Structure*.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Signature

```
ReadSafetyData (
    [in]   UInt32                InSafetyConsumerID,
    [in]   UInt32                InMonitoringNumber,
    [in]   InFlagsType           InFlags,
    [out]  Structure             OutSafetyData,
    [out]  OutFlagsType          OutFlags,
    [out]  UInt32                OutSPDU_ID_1,
    [out]  UInt32                OutSPDU_ID_2,
    [out]  UInt32                OutSPDU_ID_3,
    [out]  UInt32                OutSafetyConsumerID,
    [out]  UInt32                OutMonitoringNumber,
    [out]  UInt32                OutCRC,
    [out]  Structure             OutNonSafetyData)
;
```

Table 7 – ReadSafetyData Method arguments

Argument	Description
InSafetyConsumerID	"Safety Consumer Identifier", see *SafetyConsumerID* in Table 23.
InMonitoringNumber	"*MonitoringNumber* of the *RequestSPDU*", see 7.2.1.3 and *MonitoringNumber* in Table 23.
InFlags	"Octet with *non-safety*-related *flags* from *SafetyConsumer*", see 6.2.3.1.
OutSafetyData	"*SafetyData*", see 7.2.1.5.
OutFlags	"Octet with safety-related *flags* from *SafetyProvider*", see 6.2.3.2.
OutSPDU_ID_1	"Safety PDU Identifier Part1", see 7.2.3.2.
OutSPDU_ID_2	"Safety PDU Identifier Part2", see 7.2.3.2.
OutSPDU_ID_3	"Safety PDU Identifier Part3", see 7.2.3.2.
OutSafetyConsumerID	"Safety Consumer Identifier", see *SafetyConsumerID* in Table 23 and Table 26.
OutMonitoringNumber	*MonitoringNumber* of the *ResponseSPDU*, see 7.2.1.9, 7.2.3.1, and Figure 11.
OutCRC	*CRC* over the *ResponseSPDU*, see 7.2.3.6.
OutNonSafetyData	"Non-safe data" see 7.2.1.11.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Table 8 – ReadSafetyData Method AddressSpace definition

Attribute	Value				
BrowseName	ReadSafetyData				
References	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
HasProperty	Variable	InputArguments	Argument[]	PropertyType	Mandatory
HasProperty	Variable	OutputArguments	Argument[]	PropertyType	Mandatory
Conformance units					
ReadSafetyData					

6.2.2.4 Method ReadSafetyDiagnostics

This *Method* is mandatory for the *Facet SafetyProviderServerMapper* and optional for the *Facet SafetyProviderPubSubMapper*. It is provided for each *SafetyProvider* serving as a *Diagnostic Interface*, see 6.4.3.

See Table 9 for the arguments of *Method ReadSafetyDiagnostics* and Table 10 for its *AddressSpace* definition.

The *Method* arguments *OutSafetyData* and *OutNonSafetyData* are application-specific types derived from *Structure*.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Signature

```
ReadSafetyDiagnostics (
    [out]  UInt32                  InSafetyConsumerID,
    [out]  UInt32                  InMonitoringNumber,
    [out]  InFlagsType             InFlags,
    [out]  Structure               OutSafetyData,
    [out]  OutFlagsType            OutFlags,
    [out]  UInt32                  OutSPDU_ID_1,
    [out]  UInt32                  OutSPDU_ID_2,
    [out]  UInt32                  OutSPDU_ID_3,
    [out]  UInt32                  OutSafetyConsumerID,
    [out]  UInt32                  OutMonitoringNumber,
    [out]  UInt32                  OutCRC,
    [out]  Structure               OutNonSafetyData)
;
```

Table 9 – ReadSafetyDiagnostics Method arguments

Argument	Description
InSafetyConsumerID	see Table 7
InMonitoringNumber	see Table 7
InFlags	see Table 7
OutSafetyData	see Table 7
OutFlags	see Table 7
OutSPDU_ID_1	see Table 7
OutSPDU_ID_2	see Table 7
OutSPDU_ID_3	see Table 7
OutSafetyConsumerID	see Table 7
OutMonitoringNumber	see Table 7
OutCRC	see Table 7
OutNonSafetyData	see Table 7

Table 10 – ReadSafetyDiagnostics Method AddressSpace definition

Attribute	Value				
BrowseName	ReadSafetyDiagnostics				
References	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
HasProperty	Variable	OutputArguments	Argument[]	PropertyType	Mandatory
Conformance units					
ReadSafetyDiagnostics					

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

6.2.2.5 Object SafetyPDUs

This *Object* is mandatory for the *Facet SafetyProviderPubSubMapper* and the *Facet SafetyConsumerPubSubMapper*. It is used by the *SafetyProvider* to subscribe to the *RequestSPDU* and to publish the *ResponseSPDU*. The *DataType* of *RequestSPDU* is structured in the same way as the input arguments of *ReadSafetyData*. The *DataType* of *ResponseSPDU* is structured in the same way as the output arguments of *ReadSafetyData*.

See Table 11 for the definition of the *SafetyPDUsType*.

Both variables in the *SafetyPDUsType* have a counterpart within the *Information Model* of the *SafetyConsumer*. The *SafetyConsumer* publishes the *RequestSPDU* and subscribes to the *ResponseSPDU*.

Table 11 – SafetyPDUsType definition

Attribute	Value				
BrowseName	SafetyPDUsType				
IsAbstract	False				
References	**Node class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling rule**
Subtype of BaseObjectType					
HasComponent	Variable	<RequestSPDU>	RequestSPDUDataType	BaseDataVariableType	Mandatory Placeholder
HasComponent	Variable	<ResponseSPDU>	ResponseSPDUDataType	BaseDataVariableType	Mandatory Placeholder
Conformance units					
SafetyPDUs					

The *Object SafetyPDUs* shall contain exactly one *Reference* to a *Variable* of *DataType RequestSPDUDataType* and exactly one *Reference* to a *Variable* of a subtype of *DataType ResponseSPDUDataType*.

For example, Figure 5 shows a distributed safety application with four *SafetyAutomationComponents*. It is assumed that *SafetyAutomationComponent 1* sends a value to the other three *SafetyAutomationComponents* using three *SafetyProviders*, each comprising a pair of *SPDUs*. For each recipient, there is an individual pair of *SPDUs*.

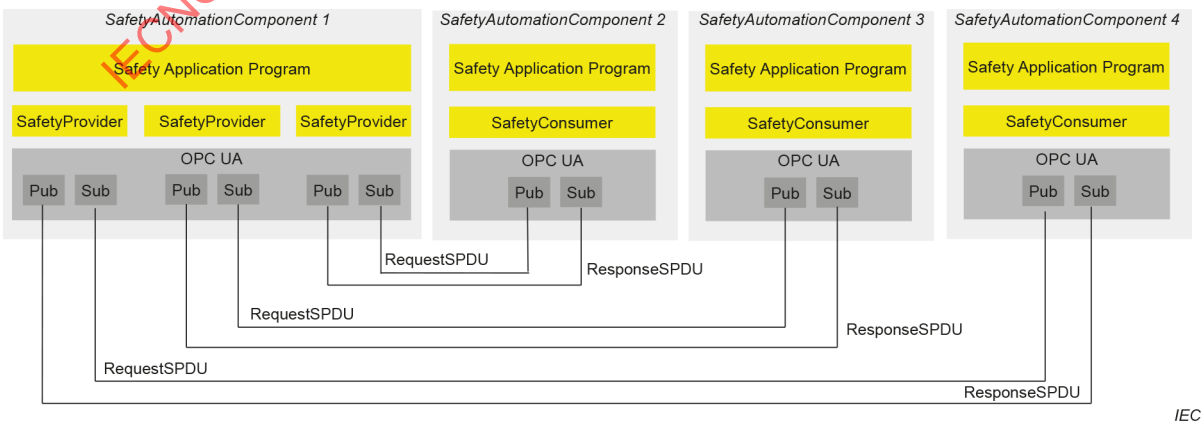
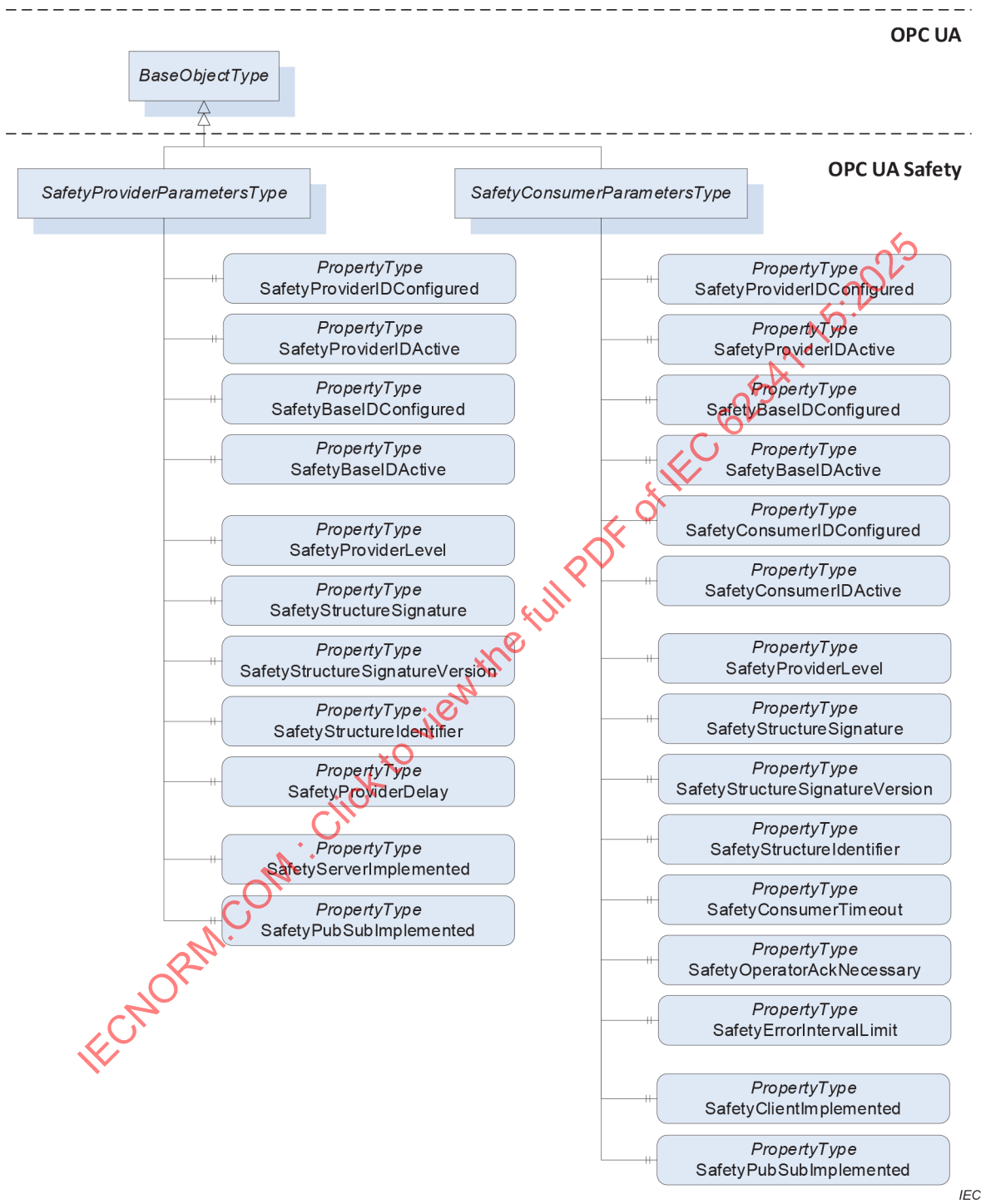


Figure 5 – Safety multicast with three recipients using IEC 62541 PubSub

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

6.2.2.6 Objects SafetyProviderParameters and SafetyConsumerParameters

Figure 6 shows the safety parameters for the *SafetyProvider* and the *SafetyConsumer*.



IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Figure 6 – Safety parameters for the SafetyProvider and the SafetyConsumer

Table 12 shows the definition for the *SafetyProviderParametersType*. Refer to 6.3.3.3 for more details on the *Safety Parameter Interface* (*SPI*) of the *SafetyProvider*.

Table 12 – SafetyProviderParametersType definition

Attribute	Value				
BrowseName	SafetyProviderParametersType				
IsAbstract	False				
References	**Node class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling rule**
Subtype of BaseObjectType					
HasProperty	Variable	SafetyProviderIDConfigured	UInt32	PropertyType	Mandatory
HasProperty	Variable	SafetyProviderIDActive	UInt32	PropertyType	Mandatory
HasProperty	Variable	SafetyBaseIDConfigured	Guid	PropertyType	Mandatory
HasProperty	Variable	SafetyBaseIDActive	Guid	PropertyType	Mandatory
HasProperty	Variable	SafetyProviderLevel	Byte	PropertyType	Mandatory
HasProperty	Variable	SafetyStructureSignature	UInt32	PropertyType	Mandatory
HasProperty	Variable	SafetyStructureSignatureVersion	UInt16	PropertyType	Mandatory
HasProperty	Variable	SafetyStructureIdentifier	String	PropertyType	Mandatory
HasProperty	Variable	SafetyProviderDelay	UInt32	PropertyType	Mandatory
HasProperty	Variable	SafetyServerImplemented	Boolean	PropertyType	Mandatory
HasProperty	Variable	SafetyPubSubImplemented	Boolean	PropertyType	Mandatory
Conformance units					
SafetyProviderParameters					

The parameters for *SafetyProviderID* and *SafetyBaseID* exist in pairs for "Configured" and "Active" states:

- *SafetyProviderIDConfigured* and *SafetyProviderIDActive*,
- *SafetyBaseIDConfigured* and *SafetyBaseIDActive*.

The "[...]Configured" parameters shall always deliver the values as configured via the *SPI*. The "[...]Active" parameters shall deliver:

- the corresponding "[...]Configured" values if the system is still offline;
- the values which have been set during runtime via the *SAPI* parameters (*SafetyProviderID*, *SafetyBaseID*);
- the corresponding "[...]Configured" values if the active values have been set to zero via the *SAPI* parameters (*SafetyProviderID*, *SafetyBaseID*).

The *Property SafetyBaseIDConfigured* is shared for all *SafetyProviders* with the same *SafetyBaseIDConfigured* value. If multiple instances of *SafetyObjectsType* are running on the same *Node*, it is a viable optimization that a *Property SafetyBaseIDConfigured* is referenced by either multiple *SafetyProviders* or *SafetyConsumers*, or both.

For releases up to Release 2.0 of the document, the value for the *SafetyStructureSignatureVersion* shall be 0x0001 (see RQ7.21 in 7.2.3.5).

Table 13 shows the definition of the *SafetyConsumerParametersType*. The *Properties SafetyStructureIdentifier* and *SafetyStructureSignatureVersion* are optional, because *SafetyStructureSignature* is typically calculated in an offline engineering tool. For small devices, it could be beneficial to only upload the *SafetyStructureSignature* to the device, but not *SafetyStructureIdentifier* and *SafetyStructureSignatureVersion* in order to save either bandwidth or memory, or both. Refer to 6.3.4.4 for more details on the *Safety Parameter Interface* (*SPI*) of the *SafetyConsumer*.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Table 13 – SafetyConsumerParametersType definition

Attribute	Value				
BrowseName	SafetyConsumerParametersType				
IsAbstract	False				
References	**Node class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling rule**
Subtype of BaseObjectType					
HasProperty	Variable	SafetyProviderIDConfigured	UInt32	PropertyType	Mandatory
HasProperty	Variable	SafetyProviderIDActive	UInt32	PropertyType	Mandatory
HasProperty	Variable	SafetyBaseIDConfigured	Guid	PropertyType	Mandatory
HasProperty	Variable	SafetyBaseIDActive	Guid	PropertyType	Mandatory
HasProperty	Variable	SafetyConsumerIDConfigured	UInt32	PropertyType	Mandatory
HasProperty	Variable	SafetyConsumerIDActive	UInt32	PropertyType	Mandatory
HasProperty	Variable	SafetyProviderLevel	Byte	PropertyType	Mandatory
HasProperty	Variable	SafetyStructureSignature	UInt32	PropertyType	Mandatory
HasProperty	Variable	SafetyStructureSignatureVersion	UInt16	PropertyType	Optional
HasProperty	Variable	SafetyStructureIdentifier	String	PropertyType	Optional
HasProperty	Variable	SafetyConsumerTimeout	UInt32	PropertyType	Mandatory
HasProperty	Variable	SafetyOperatorAckNecessary	Boolean	PropertyType	Mandatory
HasProperty	Variable	SafetyErrorIntervalLimit	UInt16	PropertyType	Mandatory
HasProperty	Variable	SafetyClientImplemented	Boolean	PropertyType	Mandatory
HasProperty	Variable	SafetyPubSubImplemented	Boolean	PropertyType	Mandatory
Conformance units					
SafetyConsumerParameters					

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

The parameters for *SafetyProviderID*, *SafetyBaseID* and *SafetyConsumerID* exist in pairs for "Configured" and "Active" states: *SafetyProviderIDConfigured* and *SafetyProviderIDActive*, *SafetyBaseIDConfigured* and *SafetyBaseIDActive*, and *SafetyConsumerIDConfigured* and *SafetyConsumerIDActive*.

The "[...]Configured" parameters shall always deliver the values as configured via the *SPI*. The "[...]Active" parameters shall deliver:

- the corresponding "[...]Configured" values if the system is still offline;
- the values which have been set during runtime via the *SAPI* parameters (*SafetyProviderID*, *SafetyBaseID*, *SafetyConsumerID*);
- the corresponding "[...]Configured" values if the active values have been set to zero via the *SAPI* parameters (*SafetyProviderID*, *SafetyBaseID*, *SafetyConsumerID*).

6.2.3 DataType definition

6.2.3.1 InFlagsType

The InFlagsType a subtype of the *Byte DataType* with the *OptionSetValues Property* defined. The *InFlagsType* is formally defined in Table 14.

CommunicationError can be used as a trigger, e.g. for a communication analysis tool. It is not forwarded to the safety application by the *SafetyProvider*. If CommunicationError is necessary in the safety application, bidirectional communication can be implemented and the value of CommunicationError can be put into the user data.

Table 14 – InFlagsType values

Value	Bit no.	Description
CommunicationError	0	0: No error 1: An error was detected in the previous *ResponseSPDU*.
OperatorAckRequested	1	Used to inform the *SafetyProvider* that operator acknowledgment is requested.
FSV_Activated	2	Is used for conformance test of SafetyConsumer.SAPI.FSV_Activated

Bits 3 to 7 are reserved for future use and shall be set to zero by the *SafetyConsumer*. They shall not be evaluated by the *SafetyProvider*.

The *InFlagsType* representation in the *AddressSpace* is defined in Table 15.

Table 15 – InFlagsType definition

Attribute		Value			
BrowseName		InFlagsType			
IsAbstract		False			
References	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of the *Byte DataType* defined in IEC 62541-3.					
0:HasProperty	Variable	0:OptionSetValues	0:LocalizedText []	0:PropertyType	
Conformance units					
SafetySupport					

6.2.3.2 OutFlagsType

The *OutFlagsType* is a subtype of the *Byte DataType* with the *OptionSetValues Property* defined. The *OutFlagsType* is formally defined in Table 16.

Table 16 – OutFlagsType values

Value	Bit no.	Description
OperatorAckProvider	0	Operator acknowledgment at the provider, hereby forwarded to the *SafetyConsumer*, see OperatorAckProvider in the *SAPI* of the *SafetyProvider*, 6.3.3.2.
ActivateFSV	1	Activation of *fail-safe* values by the safety application at the *SafetyProvider*, hereby forwarded to the *SafetyConsumer*, see *ActivateFSV* in the *SAPI* of the *SafetyProvider*, 6.3.3.2
TestModeActivated	2	*Enabling and disabling of test mode in the SafetyProvider,* hereby forwarded to the *SafetyConsumer*, *see EnableTestMode* in the *SAPI* of the *SafetyProvider*, 6.3.3.2

Bits 3 to 7 are reserved for future use and shall be set to zero by the *SafetyProvider*. They shall not be evaluated by the *SafetyConsumer*.

The *OutFlagsType* representation in the *AddressSpace* is defined in Table 17.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Table 17 – OutFlagsType definition

Attribute		Value			
BrowseName		OutFlagsType			
IsAbstract		False			
References	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**Other**
Subtype of the *Byte DataType* defined in IEC 62541-3.					
0:HasProperty	Variable	0:OptionSetValues	0:LocalizedText []	0:PropertyType	
Conformance units					
SafetySupport					

6.2.3.3 RequestSPDUDataType

Table 18 shows the definition of the *RequestSPDUDataType*. The Prefix "In" is interpreted from the *SafetyProvider*'s point of view and is used in a consistent manner to the parameters of the *Method ReadSafetyData* (see 6.2.2.3).

Table 18 – RequestSPDUDataType structure

Name	Type	Description
RequestSPDUDataType	structure	
InSafetyConsumerID	UInt32	See corresponding *Method* argument in Table 7.
InMonitoringNumber	UInt32	See corresponding *Method* argument in Table 7.
InFlags	InFlagsType	See corresponding *Method* argument in Table 7.

The representation in the *AddressSpace* of the *RequestSPDUDataType* is defined in Table 19.

Table 19 – RequestSPDUDataType definition

Attributes	Value				
BrowseName	**RequestSPDUDataType**				
IsAbstract	False				
References	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of *Structure* defined in IEC 62541-3.					
Conformance units					
SafetyPDUs					

6.2.3.4 ResponseSPDUDataType

Table 20 shows the *ResponseSPDUDataType Structure*. The Prefix "Out" is interpreted from the *SafetyProvider*'s point of view and is used in a consistent manner to the parameters of the *Method ReadSafetyData* (see 6.2.2.3).

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Table 20 – ResponseSPDUDataType structure

Name	Type	Description
ResponseSPDUDataType	structure	
OutFlags	OutFlagsType	See corresponding *Method* argument in Table 7.
OutSPDU_ID_1	UInt32	See corresponding *Method* argument in Table 7.
OutSPDU_ID_2	UInt32	See corresponding *Method* argument in Table 7.
OutSPDU_ID_3	UInt32	See corresponding *Method* argument in Table 7.
OutSafetyConsumerID	UInt32	See corresponding *Method* argument in Table 7.
OutMonitoringNumber	UInt32	See corresponding *Method* argument in Table 7.
OutCRC	UInt32	See corresponding *Method* argument in Table 7.

[RQ6.7] To define the concrete *DataType* for the *ResponseSPDU* (which specifies the concrete *DataTypes* for *SafetyData* and *NonSafetyData*, respectively), proceed as follows: (1) Derive a concrete *DataType* from the abstract *ResponseSPDUDataType*. (2) In doing so, add the following fields to the *Structure* in the given order: (a) First, field *OutSafetyData* with the concrete *Structure DataType* for the *SafetyData* (see 7.2.1.5). (b) Second, field *NonSafetyData* with the concrete *Structure DataType* for the *NonSafetyData* (or a placeholder *DataType*, see requirement RQ6.8).

[RQ6.8] To avoid possible problems with empty *Structures*, the dummy *Structure NonSafetyDataPlaceholder* shall be used as *DataType* for *OutNonSafetyData* when no *NonSafetyData* is used. The *DataType Node* defining this *Structure* has a fixed *NodeID* and contains a single *Boolean*.

The representation in the *AddressSpace* of the *ResponseSPDUDataType* is defined in Table 21.

Table 21 – ResponseSPDUDataType definition

Attributes	Value				
BrowseName	ResponseSPDUDataType				
IsAbstract	True				
References	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Subtype of *Structure* defined in IEC 62541-3.					
Conformance units					
SafetyPDUs					

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

6.2.3.5 NonSafetyDataPlaceholderDataType

Table 22 shows the definition of the *NonSafetyDataPlaceholderDataType*. The receiver shall not evaluate the value of 'dummy'.

Table 22 – NonSafetyDataPlaceholderDataType structure

Name	Type	Description
NonSafetyDataPlaceholderDataType	Structure	
Dummy	Boolean	Dummy *Variable* to avoid empty structures.

6.2.4 SafetyProvider version

Future versions may use different identifiers (such as *ReadSafetyDataV2* for the *Method* when using *Client/Server* communication or *RequestSPDUV2DataType* and *ResponseSPDUV2DataType* for the *SPDU DataTypes* when using *PubSub* communication), allowing a *SafetyProvider* to implement multiple versions of this document at the same time. Hence, the same *SafetyProvider* can be accessed by *SafetyConsumers* of different versions.

6.2.5 DataTypes and length of SafetyData

This document supports sending of the *Built-in* and *Simple DataTypes* specified in the IEC 62541 series (see IEC 62541-3and IEC 62541-6) within *SafetyData*. The supported *DataTypes* are vendor-specific.

[RQ6.9] Only scalar *DataTypes* shall be used. Arrays are currently not supported by this document.

The supported maximum length of the *SafetyData* is vendor-specific but still limited to 1 500 octets. Typical values for the maximum length include 1 octet, 16 octets, 64 octets, 256 octets, 1 024 octets, and 1 500 octets.

[RQ6.10] For controller-like devices, the supported *DataTypes* and the maximum length of the *SafetyData* shall be listed in the user manual.

[RQ6.11] For the *DataType Boolean*, the value 0x01 shall be used for 'true' and the value 0x00 shall be used for 'false'.

It is recommended to send multiple *Booleans* in separate variables. However, in small devices, it can be necessary to combine a set of 8 *Booleans* in one *Variable* for performance reasons. In this case, the *DataType Byte* can be used.

6.2.6 Connection establishment

This document uses the IEC 62541 services for connection establishment, it poses no additional requirement to these services.

This version of the document describes configuration only at engineering time. This means that the parameters defined in the *SPI* (see 6.3.3.3 and 6.3.4.4) are read-only via the interface described in this document. Changing of parameters is expected to be done in a safety-related way, using the respective tools and interfaces provided by the vendor. Future versions of this document may specify a vendor-independent interface for configuration.

6.3 Service interfaces

6.3.1 Overview

Figure 7 gives an overview of the *safety communication layer* and its interfaces. It thereby also shows the scope of this document. The main function of the layer services is the state machine which handles the protocol. The state machines interact with the following interfaces:

– The *Safety Application Program Interface* (*SAPI*) is accessed by the safety application for exchanging *SafetyData* during runtime.

– The *Safety Parameter Interface* (*SPI*) is accessed during commissioning for setting safety parameters such as IDs or the timeout value in the *SafetyConsumer*.

– The *non-safety* related *Diagnostic Interface* (*DI*) can be accessed at runtime for troubleshooting the safety communication.

– The *OPC UA Platform Interface* (*OPC UA PI*) connects the *SCL* to the *non-safe* IEC 62541 stack and is used during runtime.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

The interfaces (*SAPI*, *SPI*, *DI* and *OPC UA PI*) described in 6.3 are abstract and informative. They represent logical data inputs and outputs to this layer that are necessary for the proper operation of the state machine. No normative, concrete mappings are specified. The concrete implementations are vendor-specific and do no necessarily exactly match the abstract interfaces described in this document.

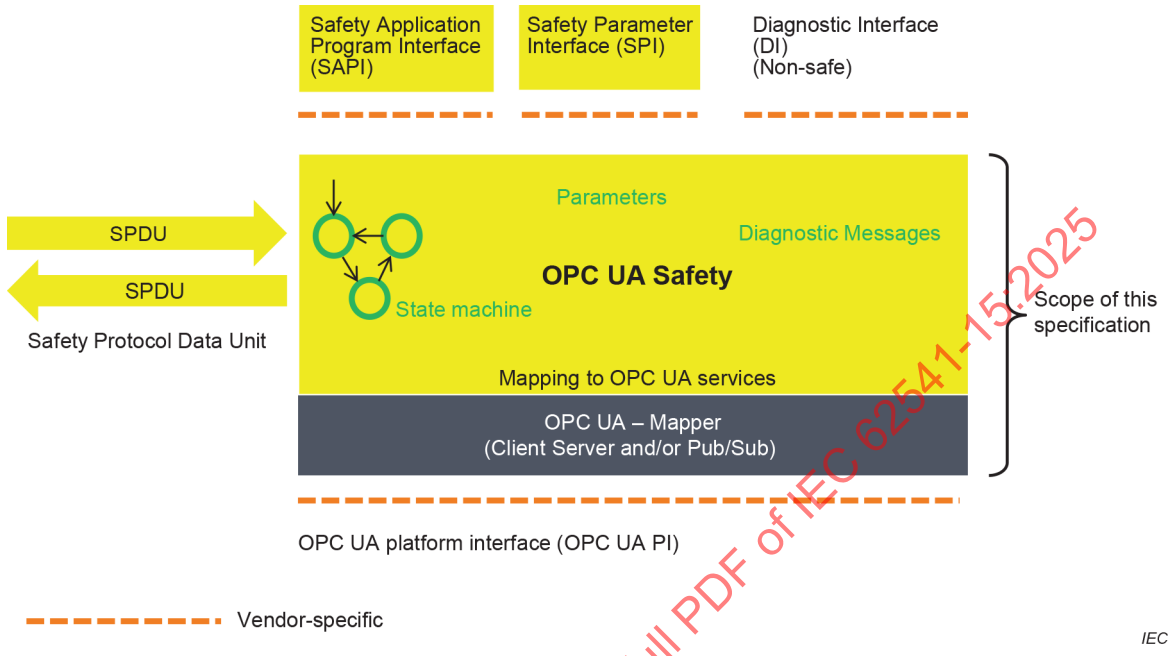


Figure 7 – Safety communication layer overview

6.3.2 OPC UA Platform interface (OPC UA PI)

The state machines of this document are independent from the actual IEC 62541 services used for data transmission. This is accomplished by introducing a so-called *OPC UA mapper*, serving as an interface between the *safety communication layer* and the IEC 62541 stack.

The mapper can either make use of IEC 62541 *Client/Server* and remote *Method* invocation or the publishing of and subscribing to remote *Variables* as defined in IEC 62541-14. The requirements on the implementation of the mapper are implicitly given in 6.2.

6.3.3 SafetyProvider interfaces

6.3.3.1 General

Figure 8 shows an overview of the *SafetyProvider* interfaces. The *SAPI* is specified in 6.3.3.2, the *SPI* is specified in 6.3.3.3.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

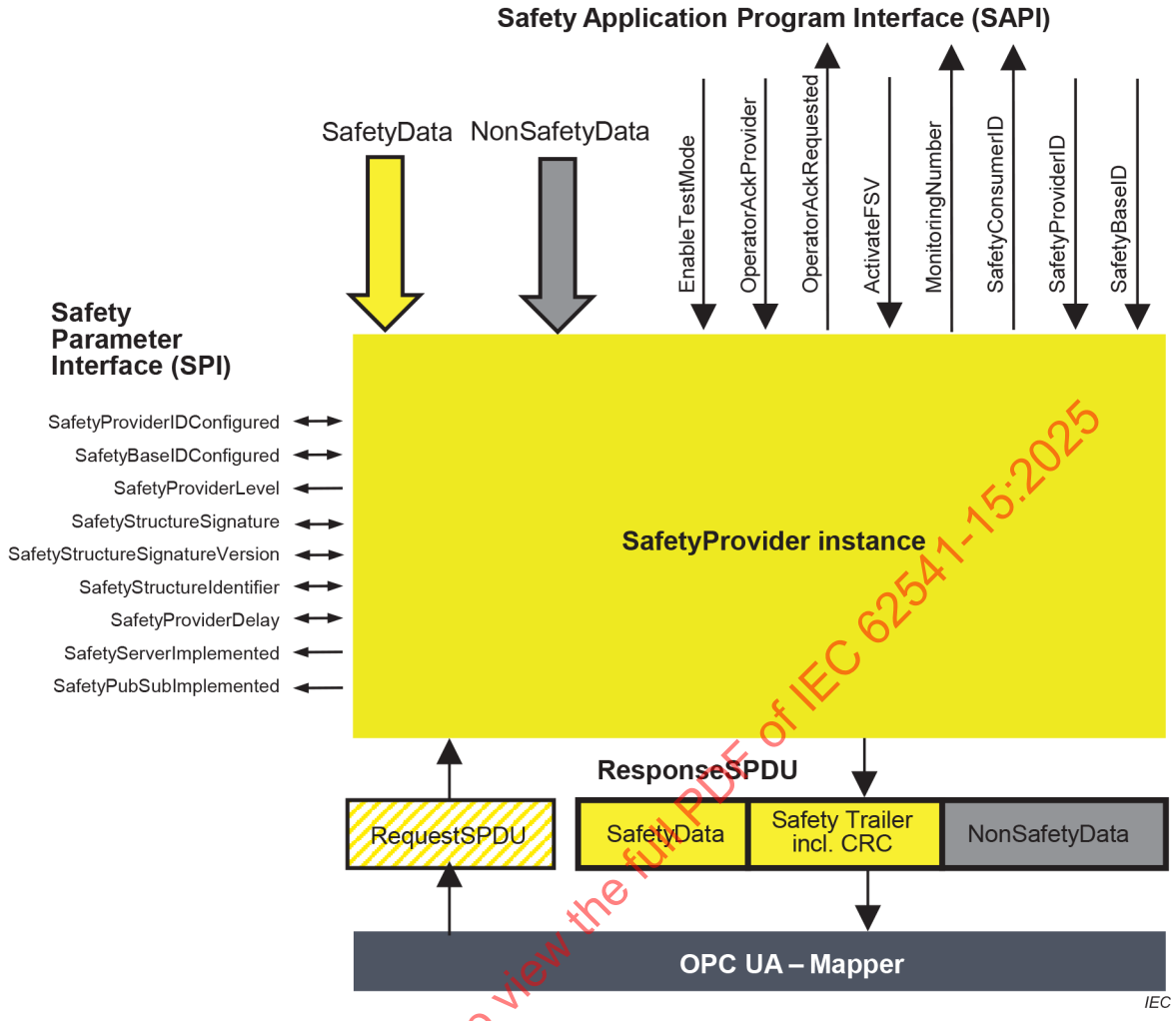


Figure 8 – SafetyProvider interfaces

6.3.3.2 SAPI of SafetyProvider

[RQ6.12] The *SAPI* of the *SafetyProvider* represents the *safety communication layer* services of the *SafetyProvider*. Table 23 lists all inputs and outputs of the *SAPI* of the *SafetyProvider*. Each *SafetyProvider* shall implement the *SAPI* as shown in Table 23, however, the details are vendor-specific.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Table 23 – SAPI of the SafetyProvider

SAPI Term	Type	I/O	Definition
SafetyData	Structure	I	This input is used to accept the user data which is then transmitted as *SafetyData* in the *SPDU*. NOTE 1 Whenever a new *MNR* is received from a *SafetyConsumer*, the state machine of the *SafetyProvider* will read a new value of the *SafetyData* from its corresponding Safety Application and use it until the next *MNR* is received. If no valid user data is available at the Safety Application, ActivateFSV can be set to "1" by the Safety Application.
NonSafetyData	Structure	I	Used to consistently transmit *NonSafetyData* values (e.g. diagnostic information) together with safe data, see 7.2.1.11
EnableTestMode	Boolean	I	By setting this input to "1" the **remote** *SafetyConsumer* is informed (by Bit 2 in *ResponseSPDU.Flags*, see 6.2.3.2) that the *SafetyData* are test data, and are not to be used for safety-related decisions. NOTE 2 This document is intended for implementation in safety devices exclusively, see requirement RQ4.1.
OperatorAckProvider	Boolean	I	This input is used to implement an operator acknowledgment on the *SafetyProvider* side. The value will be forwarded to the *SafetyConsumer*, where it can be used to trigger a return from *fail-safe* substitute values (*FSV*) to actual *process values* (*PV*), see B.3.4.
OperatorAckRequested	Boolean	O	Indicates that an operator acknowledge is requested by the *SafetyConsumer*. This *flag* is received within the *RequestSPDU*.
ActivateFSV (*Fail-safe* substitute values)	Boolean	I	By setting this input to "1" the *SafetyConsumer* is instructed (via Bit 1 in *ResponseSPDU.Flags*, see 6.2.3.2) to deliver *FSV* instead of *PV* to the safety application program. NOTE 3 If the replacement of *process values* by *FSV* is to be controllable in a more fine-grained way, this can be realized by using *qualifiers* within the *SafetyData*, see 6.3.6.
SafetyConsumerID	UInt32	O	This output yields the ConsumerID used in the last access to this *SafetyProvider* by a *SafetyConsumer* (see 6.2.2.3). Since all safety-related checks are executed by an implementation of this document, the safety application is not required to check this *SafetyConsumerID*.
MonitoringNumber	UInt32	O	This output yields the *MonitoringnNumber* (*MNR*). It is updated whenever a new request comes in from the *SafetyConsumer*. Since all safety-related checks are executed by an implementation of this document, the safety application is not required to check this *MonitoringNumber*.
SafetyProviderID	UInt32	I	For dynamic systems, this input can be set to a non-zero value. In this case, the *SafetyProvider* uses this value instead of the value from the *SPI* parameter *SafetyProviderIDConfigured*. If the value is changed to "0", the value of parameter *SafetyProviderIDConfigured* from the *SPI* will be used (again). See Figure 8, 3.1.2.15, and 9.1.1. For static systems, this input is usually always kept at value "0".
SafetyBaseID	Guid	I	For dynamic systems, this input can be set to a non-zero value. In this case, the *SafetyProvider* uses this value instead of the value of the *SPI* parameter *SafetyBaseIDConfigured*. If the value is changed to "0", the value of parameter *SafetyBaseIDConfigured* from the *SPI* will be used (again). See Figure 8, 3.1.2.14, and 9.1.1. For static systems, this input is usually always kept at value "0".

For additional information regarding operator acknowledgment use cases, see Clause B.3.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

6.3.3.3 SPI of SafetyProvider

[RQ6.13a] Each *SafetyProvider* shall implement the parameters and constants [RQ6.13b] as shown in Table 24. The parameters (R/W in column "Access") can be set via the *SPI*, whereas the constants (R in column "Access") are read-only. The mechanisms for setting the parameters are vendor-specific. The attempt of setting a parameter to a value outside its range, or of the setting of a read-only parameter, shall not become effective, and a diagnostic message should be shown when appropriate. The values of the constants depend on the way the *SafetyProvider* is implemented. They never change and are therefore not writable via any of the interfaces.

Table 24 – SPI of the SafetyProvider

Identifier	Type	Range	Initial value (before configuration)	Access	Note
SafetyProviderIDConfigured	UInt32	0x0 to 0xFFFFFFFF	0x0	R/W	*SafetyProviderID* of the *SafetyProvider* that is normally used, see 3.1.2.13 and 3.1.2.15 and 9.1.1. For dynamic systems, the safety application program can overwrite this ID by providing a non-zero value at the input *SafetyProviderID* of the *SafetyProvider*'s *SAPI*. This runtime value can be queried using the *SafetyProviderIDActive* parameter. See 6.2.2.6 for details on configured and active values. NOTE 1 If both the values provided at the *SPI* and the *SAPI* are 0x0, this means that the *SafetyProvider* is not properly configured. *SafetyConsumers* will never try to communicate with *SafetyProviders* having a *SafetyProviderID* of 0x0, see Transitions T13/T27 in Table 35 and the macro <ParametersOK?> in Table 33.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Identifier	Type	Range	Initial value (before configuration)	Access	Note
SafetyBaseIDConfigured	Guid	Any value which can be repre-sented with sixteen octets	All sixteen octets are 0x00	R/W	*SafetyBaseID* of the *SafetyProvider* that is normally used, see 3.1.2.14 and 3.1.2.13 and 9.1.1. For dynamic systems, the safety application program can overwrite this ID by providing a non-zero value at the input *SafetyBaseID* of the *SafetyProvider*'s *SAPI*. This runtime value can be queried using the *SafetyBaseIDActive* parameter. See 6.2.2.6 for details on configured and active values. NOTE 2 If both the values provided at the *SPI* and the *SAPI* are 0x0, this means that the *SafetyProvider* is not properly configured. *SafetyConsumers* will never try to communicate with *SafetyProviders* having a *SafetyBaseID* of 0x0, see Transitions T13/T27 in Table 35 and the macro <ParametersOK?> in Table 33. See 9.1.1 for more information on *GUID*.
SafetyProviderLevel	Byte	0x01 to 0x04	n.a.	R	The *SIL* the *SafetyProvider* implementation (hardware and software) is capable of, see Figure 9. NOTE 3 It is independent from the generation of the *SafetyData* at *SAPI*. NOTE 4 The *SafetyProviderLevel* is used to distinguish devices of a different *SIL*. As a result, *SPDUs* coming from a device with a low *SIL* will never be accepted when a *SafetyConsumer* is parameterized to implement a safety function with a high *SIL*.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Identifier	Type	Range	Initial value (before configuration)	Access	Note
SafetyStructureSignature	UInt32	0x0 to 0xFFFF FFFF	0x0	R/W	Signature of the *SafetyData* structure, for calculation see 7.2.3.5. NOTE 5 "0" would not be a valid signature and thus indicates a *SafetyProvider* which is not properly configured. *SafetyConsumers* will never try to communicate with *SafetyProviders* having a *SafetyStructureSignature* of 0x0, see Transitions T13/T27 in Table 35 and the macro <ParametersOK?> in Table 33.
SafetyStructureSignature Version	UInt16	0x1	0x1	R/W	Version used to calculate *SafetyStructureSignature*, see 7.2.3.5
SafetyStructureIdentifier	String	all strings	"" (the empty string)	R/W	Identifier describing the *DataType* of the *SafetyData*, see 7.2.3.5.
SafetyProviderDelay	UInt32	0x0 to 0xFFFF FFFF	0x0	R/W	In microseconds (µs). It can be set during the engineering phase of the *SafetyProvider* or set during online configuration as well. *SafetyProviderDelay* is the maximum time at the *SafetyProvider* from receiving the *RequestSPDU* to start the transmission of *ResponseSPDU*, see 8.1. The parameter *SafetyProviderDelay* has no influence on the functional behaviour of the *SafetyProvider*. Therefore, it is not necessary for this value to be generated in a safety-related way. However, it will be provided in the IEC 62541 *Information Model* of a *SafetyProvider* to inform about its worst-case delay time. The value can be used during commissioning to check whether the timing behaviour of the *SafetyProvider* is suitable to fulfill the watchdog delay of the corresponding *SafetyConsumer*.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Identifier	Type	Range	Initial value (before configuration)	Access	Note
SafetyServerImplemented	Boolean	0x0 or 0x1	n.a.	R	This read-only parameter indicates whether the *SafetyProvider* has implemented the *Server* part of IEC 62541 *Client/Server* communication (see 5.4): 1: *Server* for IEC 62541 *Client/Server* communication is implemented. 0: *Server* for IEC 62541 *Client/Server* communication is not implemented. The corresponding *Facets* are *SafetyProviderServer* and *SafetyProviderServer-Mapper*.
SafetyPubSubImplemented	Boolean	0x0 or 0x1	n.a.	R	This read-only parameter indicates whether the *SafetyProvider* has implemented the necessary publishers and subscribers for IEC 62541 *PubSub* communication (see 5.4): 1: IEC 62541 *PubSub* communication is implemented. 0: IEC 62541 *PubSub* communication is not implemented. The corresponding *Facets* are *SafetyProviderPubSub* and *SafetyProviderPubSub-Mapper*.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

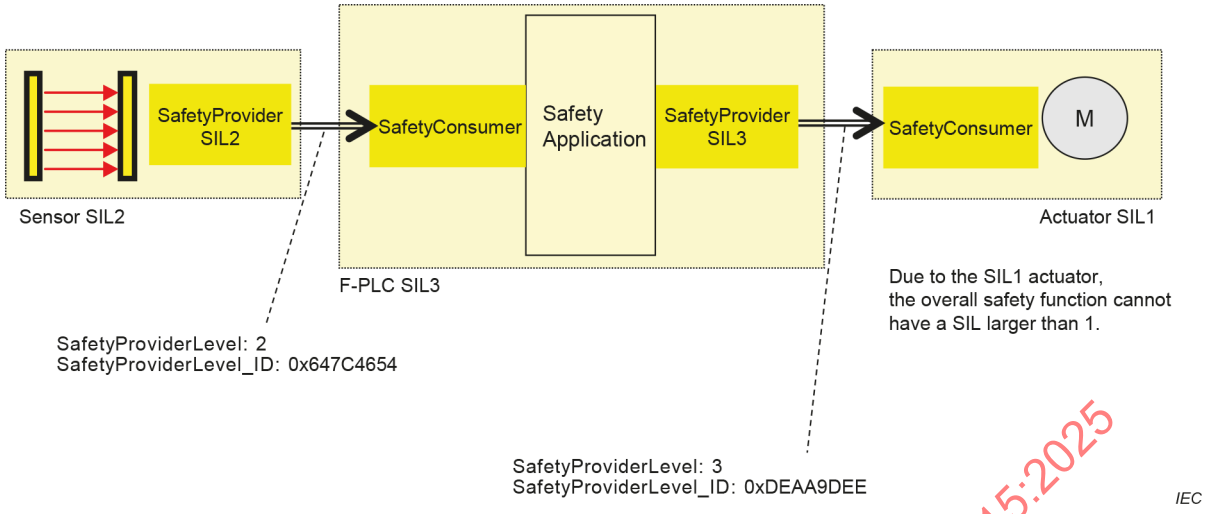


Figure 9 – Example combinations of SIL capabilities

The constant *SafetyProviderLevel* determines the value that is used for *SafetyProviderLevel_ID* when calculating the *SPDU_ID*, see 7.2.3.3.

NOTE *SafetyProviderLevel* is defined as the *SIL* the *SafetyProvider* implementation (hardware and software) is capable of, not to be confused with the *SIL* of the implemented safety function. For instance, Figure 9 shows a safety function which is implemented using a *SIL*2-capable sensor, a *SIL*3-capable PLC, and a *SIL*1-capable actuator. The overall *SIL* of the safety function is considered to be *SIL*1. Nevertheless, the *SafetyProvider* implemented on the sensor will use the constant value "2" as *SafetyProviderLevel*, whereas the *SafetyProvider* implemented on the PLC will use the constant value "3" as *SafetyProviderLevel*.

It is necessary for the respective *SafetyConsumers* (on the PLC and the actuator) to know the *SafetyProviderLevel* of their *SafetyProviders* in order to check the *SPDU_ID* (see 7.2.3.2).

6.3.4 SafetyConsumer interfaces

6.3.4.1 General

Figure 10 shows an overview of the *SafetyConsumer* interfaces. The *Safety Application Program Interface* (*SAPI*) is specified in 6.3.4.2, the *Safety Parameter Interface* (*SPI*) is specified in 6.3.4.4.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

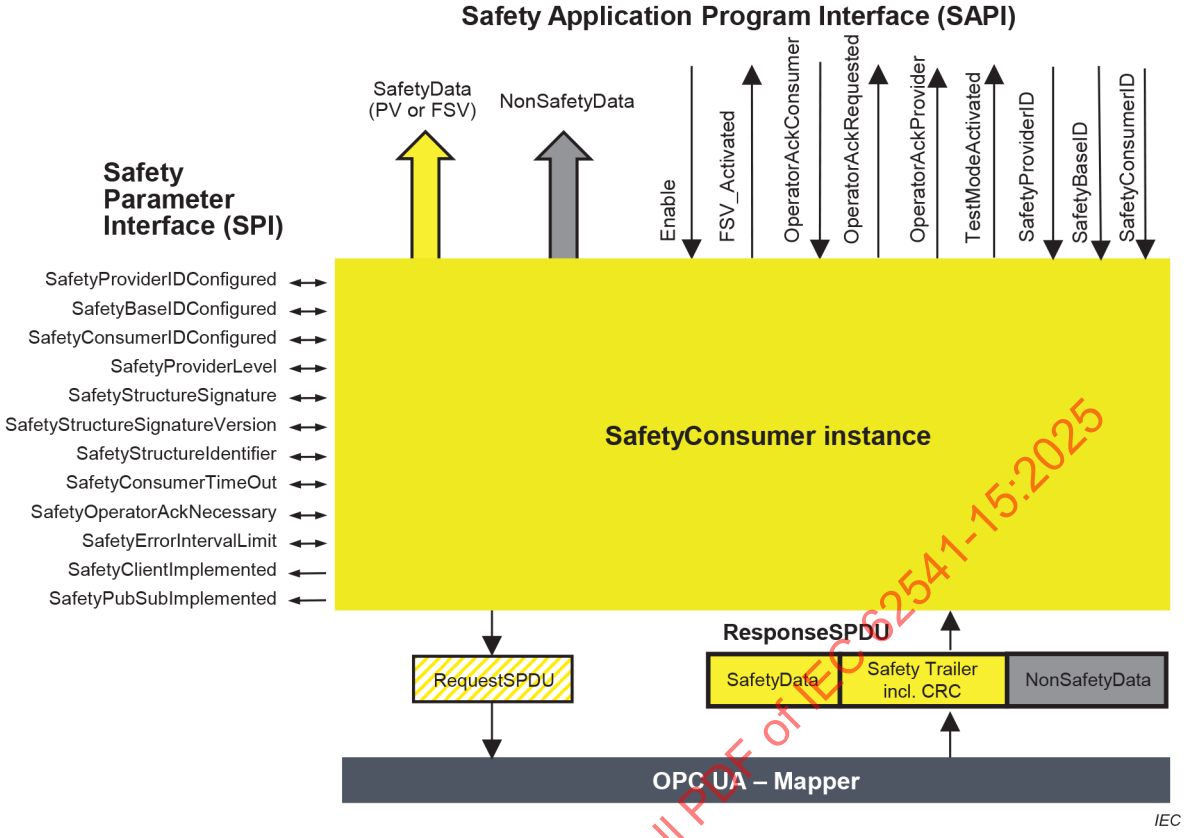


Figure 10 – SafetyConsumer interfaces

6.3.4.2 SAPI of SafetyConsumer

The *SAPI* of the *SafetyConsumer* represents the *safety communication layer* services of the *SafetyConsumer*. Table 25 lists all inputs and outputs of the *SAPI* of the *SafetyConsumer*.

[RQ6.14] Each *SafetyConsumer* shall implement the *SAPI* as shown in Table 25, however, the details are vendor-specific.

Table 25 – SAPI of the SafetyConsumer

SAPI Term	Type	I/O	Definition
SafetyData	Structure	O	This output either delivers the *process values* received from the *SafetyProvider* in the *SPDU* field *SafetyData*, or *FSV*.
NonSafetyData	Structure	O	This output delivers the *non-safety process values* (e.g. diagnostic information) which were sent together with safe data, see 7.2.1.11
Enable	Boolean	I	By changing this input to "0" the *SafetyConsumer* will change each and every *Variable* of the *SafetyData* to "0"[1] and stop sending requests to the *SafetyProvider*. When changing *Enable* to "1" the *SafetyConsumer* will restart safe communication. The variable can be used to delay the start of the safety communication according to this document, after power on until "IEC 62541 connection ready" is set.The delay time is not monitored while enable is set to "0".
FSV_Activated	Boolean	O	This output indicates via "1", that on the output *SafetyData FSV* (all binary "0") are provided[1]. NOTE 1 A *ResponseSPDU* which is checked with an error results to FSV_Activated being set to "1", see T24 in Table 35. There could be other reasons.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

SAPI Term	Type	I/O	Definition
OperatorAckConsumer	Boolean	I	For motivation, see 6.3.4.3. After an indication of OperatorAckRequested this input can be used to signal an operator acknowledgment. By changing this input from "0" to "1" (rising edge) the *SafetyConsumer* is instructed to switch *SafetyData* from *FSV* to *PV*. OperatorAckConsumer is processed only if this rising edge arrives after OperatorAckRequested was set to "1", see Figure 19. If a rising edge of OperatorAckConsumer arrives before OperatorAckRequested becomes 1, this rising edge is ignored.
OperatorAckRequested	Boolean	O	This output indicates the request for operator acknowledgment. The bit is set to "1" by the *SafetyConsumer*, when three conditions are met: 1) Too many communication errors were detected in the past, so the *SafetyConsumer* decided to switch to *fail-safe* substitute values. 2) Currently, no communication errors occur, and hence operator acknowledgment is possible. 3) Operator acknowledgment (rising edge at input OperatorAckConsumer) has not yet occurred. The bit is reset to "0" when a rising edge at OperatorAckConsumer is detected.
OperatorAckProvider	Boolean	O	This output indicates that an operator acknowledgment has taken place on the *SafetyProvider*. If operator acknowledgment at the *SafetyProvider* should be allowed, this output is connected to OperatorAckConsumer, see B.3.4 and B.3.5. NOTE 2 If the *ResponseSPDU* is checked with error, this output remains at its last value, see T24 in Table 35.
TestModeActivated	Boolean	O	The safety application program is expected to evaluate this output for determining whether the communication partner is in test mode or not. A value of "1" indicates that the communication partner (source of data) is in test mode, e.g. during commissioning. Data coming from a device in test mode may be used for testing but is not intended to be used to control safety-critical processes. A value of "0" represents the "normal" safety-related mode. The test mode enables the programmer and commissioner to validate the safety application using test data. NOTE 3 If the *ResponseSPDU* check results in an error and the *ErrorIntervalTimer* (see 6.3.4.4) is also not expired, TestModeActivated is reset, see state S17_Error in Table 34.
SafetyProviderID	UInt32	I	For dynamic systems, this input can be set to a non-zero value. In this case, the *SafetyConsumer* uses this variable instead of the *SPI* parameter *SafetyProviderIDConfigured*. This input is only read in the first cycle, or when a rising edge occurs at the input Enable. See also Table 26. If it is changed to "0", the value of *SPI* parameter *SafetyProviderIDConfigured* will be used (again). For static systems, this input is usually always kept at value "0".
SafetyBaseID	Guid	I	For dynamic systems, this input can be set to a non-zero value. In this case, the *SafetyConsumer* uses this variable instead of the *SPI* parameter *SafetyBaseIDConfigured*. This input is only read in the first cycle, or when a rising edge occurs at the input Enable. See also Table 26. If it is changed to "0", the *SPI* parameter *SafetyBaseIDConfigured* will become activated. For static systems, this input is usually always kept at value "0".

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

SAPI Term	Type	I/O	Definition
SafetyConsumerID	UInt32	I	For dynamic systems, this input can be set to a non-zero value. In this case, the *SafetyConsumer* uses this variable instead of the *SPI* parameter *SafetyConsumerID*. This input is only read in the first cycle, or when a rising edge occurs at the input Enable. See also Table 26. If it is changed to "0", the *SPI* parameter *SafetyConsumerID* will become activated. For static systems, this input is usually always kept at value "0".
[1] If an application requires different *FSV* than "all binary 0", it is expected to use appropriate constants and ignore the output of *SafetyData* whenever FSV_Activated is set.			

For additional information regarding operator acknowledgment use cases, see Clause B.3.

6.3.4.3 Motivation for SAPI Operator Acknowledge (OperatorAckConsumer)

The safety argumentation assumes that random errors in the underlying IEC 62541 stack including its communication links are not too frequent, i.e. that its failure rate is lower than a given threshold, depending on the desired *SIL* (see 9.3.1).

Whenever the *SafetyConsumer* detects a faulty message, it checks whether the assumption is still valid, and switches to *fail-safe* substitute values otherwise. Returning to *process values* then requires an operator acknowledgment.

Operator Acknowledge is expected to be initiated by a human operator who is responsible to check the installation, see Table 40, row "Operator Acknowledge". For this reason, the parameter OperatorAckRequested is delivered by the *SafetyConsumer* to the safety application. See Clause B.2 for details on operator acknowledgment scenarios.

Timeout errors do only require an operator acknowledgment if operator acknowledgment is required by the safety function itself. In this case, SafetyOperatorAckNecessary is set to indicate that operator acknowledgments are required. See 6.3.4.5 for details.

6.3.4.4 SPI of the SafetyConsumer

[RQ6.15a] Each *SafetyConsumer* shall implement the parameters and constants [RQ6.15b] as shown in Table 26. The parameters (R/W in column "Access") can be set via the *SPI*, whereas the constants (R in column "Access") are read-only. The mechanisms for setting these parameters are vendor-specific. The attempt of setting a parameter to a value outside its range, or of the setting of a read-only parameter, shall not become effective, and a diagnostic message should be shown when appropriate. The *SPI* of the *SafetyConsumer* represents the parameters of the *safety communication layer* management of the *SafetyConsumer*. The values of the constants depend on the way the *SafetyConsumer* is implemented. They never change and are therefore not writable via any of the interfaces.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Table 26 – SPI of the SafetyConsumer

Identifier	Type	Valid range	Initial value (before configu-ration)	Access	Note
SafetyProviderIDConfigured	UInt32	0x0 to 0xFFFFFFFF	0x0	R/W	The default *SafetyProviderID* of the *SafetyProvider* this *SafetyConsumer* uses to make a connection, see Figure 8 and 3.1.2.15. For dynamic systems, the safety application program can overwrite this ID by providing a non-zero value at the input *SafetyProviderID* of the *SafetyConsumer*'s *SAPI*. This runtime value can be queried using the *SafetyProviderIDActive* parameter. See 6.2.2.6 for details on configured and active values.
SafetyBaseIDConfigured	Guid	Any value which can be represen-ted with sixteen octets.	All sixteen octets are 0x0	R/W	The default *SafetyBaseID* of the *SafetyProvider* this *SafetyConsumer* uses to make a connection, see 3.1.2.14. For dynamic systems, the safety application program can overwrite this ID by providing a non-zero value at the input *SafetyBaseID* of the *SafetyConsumer*'s *SAPI*. This runtime value can be queried using the *SafetyBaseIDActive* parameter. See 6.2.2.6 for details on configured and active values. See 9.1.1 for more information on *GUID*.
SafetyConsumerIDConfigured	UInt32	0x0 to 0xFFFFFFFF	0x0	R/W	*SafetyConsumerID* of the *SafetyConsumer*, see 9.1.2. For dynamic systems, the safety application program can overwrite this ID by providing a non-zero value at the input *SafetyConsumerID* of the *SafetyConsumer*'s *SAPI*. This runtime value can be queried using the *SafetyConsumerIDActive* parameter. See 6.2.2.6 for details on configured and active values.
SafetyProviderLevel	Byte	0x01 to 0x04	0x04	R/W	*SafetyConsumer*'s expectation on the *SIL* the *SafetyProvider* implementation (hardware and software) is capable of. See 7.2.3.3, and Figure 9.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Identifier	Type	Valid range	Initial value (before configu-ration)	Access	Note
SafetyStructureSignature	UInt32	0x0 to 0xFFFFFFFF	0x0	R/W	Signature over the *SafetyData* structure, see 7.2.3.5.
SafetyStructureSignatureVersion	UInt16	0x1	0x1	R/W	Version used to calculate *SafetyStructureSignature*, see 7.2.3.5. For the *SafetyConsumer*, this parameter is optional.
SafetyStructureIdentifier	String		""	R/W	Identifier describing the *DataType* of the *SafetyData*, see 7.2.3.5. For the *SafetyConsumer*, this parameter is optional.
SafetyConsumerTimeout	UInt32	0x0 to 0xFFFFFFFF	0x0	R/W	Watchdog-time in microseconds (µs). Whenever the *SafetyConsumer* sends a request to a *SafetyProvider*, its watchdog timer is set to this value. The expiration of this timer prior to receiving an error-free reply by the *SafetyProvider* indicates an unacceptable delay. See 8.1
SafetyOperatorAckNecessary	Boolean	0x0 or 0x1	0x1	R/W	This parameter controls whether an operator acknowledgment (OA) is necessary in case of errors of type "unacceptable delay" or "loss", or when the *SafetyProvider* has activated *FSV* (ActivateFSV). 1: *FSV* are provided at the output *SafetyData* of the *SAPI* until OA. 0: *PV* are provided at *SafetyData* of the *SAPI* as soon as the communication is free of errors. In case of ActivateFSV the values change from *FSV* to *PV* as soon as ActivateFSV returns to "0". NOTE This parameter does not have an influence on the behaviour of the *SafetyConsumer* following the detection of other types of communication errors, such as data corruption or an error detected by the *SPDU_ID*. For these types of errors, OA is mandatory, see 6.3.4.3.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Identifier	Type	Valid range	Initial value (before configu-ration)	Access	Note
SafetyErrorIntervalLimit	UInt16	6, 60, 600	600	R/W	Value in minutes. The parameter *SafetyErrorIntervalLimit* determines the minimal time interval between two consecutive communication errors so that they do not trigger a switch to *FSV* in the *SafetyConsumer*, see 6.3.4.3. It affects the availability and either the PFH or PFDavg, or both, of the safety communication link according to this document, see 9.4.
SafetyClientImplemented	Boolean	0x0 or 0x1	n.a.	R	This read-only parameter indicates whether the *SafetyConsumer* has implemented the client part of IEC 62541 *Client/Server* communication (see 5.4): 1: Client for IEC 62541 *Client/Server* communication is implemented. 0: Client for IEC 62541 *Client/Server* communication is not implemented. The corresponding *Facet* is *SafetyConsumerClient*.
SafetyPubSubImplemented	Boolean	0x0 or 0x1	n.a.	R	This read-only parameter indicates whether the *SafetyConsumer* has implemented the necessary publishers and subscribers for IEC 62541 *PubSub* communication (see 5.4): 1: IEC 62541 *PubSub* communication is implemented. 0: IEC 62541 *PubSub* communication is not implemented. The corresponding *Facets* are *SafetyConsumerPubSub* and *SafetyConsumerPubSub Mapper*.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

6.3.4.5 Motivation for SPI SafetyOperatorAckNecessary

This parameter determines whether automatic restart (i.e. automatically switching back from *fail-safe* values to *process values*) is possible for the safety function or not. It is expected to be set to 1 for safety functions where automatic restart is not allowed and restart always requires human interaction.

If automatic restart of the safety function is safe, the parameter can be set to 0.

6.3.5 Cyclic and acyclic safety communication

This document supports cyclic and acyclic safety communication.

For most safety functions it is necessary to react in a timely manner to external events, such as an emergency stop button being pressed or a light curtain being interrupted. In these applications, cyclic safety communication is established. That means the *SafetyConsumer* is executed cyclically, and the time between two consecutive executions is safely bounded. The maximum time between two executions of the *SafetyConsumer* will contribute to the *safety function response time* (*SFRT*).

Some safety functions, such as the transfer of safe configuration data at startup, do not have to react on external events. In this case, it is not required to execute the *SafetyConsumer* cyclically.

6.3.6 Principle for "application variables with qualifier"

Qualifiers allow the *SafetyProvider* to indicate the correctness of values on a fine-grained level. It is good practice to attach *qualifiers* to each individual value sent within an *SPDU*. The *qualifiers* are part of the *SafetyData* and hence not within the scope of this document.

[RQ6.16] However, whenever *qualifiers* are used, the values shown in Table 27 shall be used, i.e. 0x1 for a valid value ("good"), and 0x0 for an invalid value ("bad").

Table 27 – Example "application variables with qualifier"

Value	Qualifier
valid	0x1 (= good)
invalid	0x0 (= bad)

Checking of the *qualifiers* is done in the safety application.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

6.4 Diagnostics

6.4.1 General

Diagnostics according to this document may be implemented in a *non-safety*-related way. This allows for categorization and localization of safety communication errors.

This document provides two types of diagnostics:

– Diagnostics messages generated by the *SafetyConsumer* and provided in a vendor-specific way.

– The *Method ReadSafetyDiagnostics*, defined in the IEC 62541 *Information Model* (see 6.2.2.4 and 6.4.3).

6.4.2 Diagnostics messages of the SafetyConsumer

[RQ6.17] Every time the macro <Set Diag(SD_IDerrOA, isPermanent)> is executed within the *SafetyConsumer*, the textual representation shown in Table 28 shall be presented. The details and location of this representation (display, logfile, etc.) are vendor-specific.

Table 28 – Safety layer diagnostic messages

Internal identifier (as used in the state-machines)	General error type (String)	Extended error type (String)	Error code (offset)[1]	Classification *) (optional)	Mandatory
SD_IDerrIgn	The *SafetyConsumer* has discarded a message due to an incorrect ID.		0x01	A	Yes
SD_IDerrOA	The *SafetyConsumer* has switched to *fail-safe substitute values* due to an incorrect ID. Operator acknowledgment is required.	Mismatch of *SafetyBaseID*.[2]	0x11	B, E	Yes
SD_IDerrOA	The *SafetyConsumer* has switched to *fail-safe substitute values* due to an incorrect ID. Operator acknowledgment is required.	Mismatch of *SafetyProviderID*.	0x12	B, E	Yes
SD_IDerrOA	The *SafetyConsumer* has switched to *fail-safe substitute values* due to an incorrect ID. Operator acknowledgment is required.	Mismatch of *SafetyData* structure or identifier.[3]	0x13	B, E	Yes
SD_IDerrOA	The *SafetyConsumer* has switched to *fail-safe substitute values* due to an incorrect ID. Operator acknowledgment is required.	Mismatch of *SafetyProviderLevel*.[4]	0x14	B, E	Yes
CRCerrIgn	The *SafetyConsumer* has discarded a message due to a *CRC* error (data corruption).		0x05	A	Yes
CRCerrOA	The *SafetyConsumer* has switched to *fail-safe substitute values* due to a *CRC* error (data corruption). Operator acknowledgment is required.		0x15	B, C	Yes
ColDerrIgn	The *SafetyConsumer* has discarded a message due to an incorrect ConsumerID.		0x06	A	Yes
ColDerrOA	The *SafetyConsumer* has switched to *fail-safe substitute values* due to an incorrect SafetyConsumerID. Operator acknowledgment is required.		0x16	B	Yes

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Internal identifier (as used in the state-machines)	General error type (String)	Extended error type (String)	Error code (offset)[1]	Classification *) (optional)	Mandatory
MNRerrIgn	The *SafetyConsumer* has discarded a message due to an incorrect *MonitoringNumber*.		0x07	A	Yes
MNRerrOA	The *SafetyConsumer* has switched to *fail-safe substitute values* due to an incorrect monitoring number. Operator acknowledgment is required.		0x17	B, C	Yes
CommErrTO	The *SafetyConsumer* has switched to *fail-safe substitute values* due to timeout.		0x08	B	Yes
ApplErrTO	The *SafetyConsumer* has switched to *fail-safe substitute values* at the request of the safety application.		0x09	D	No
ParametersInvalid	The *SafetyConsumer* has been configured with invalid parameters.		0x0A	B, E	Yes
FSV_Requested	The *SafetyConsumer* has switched to *fail-safe substitute values* at the request of the *SafetyProvider*. Operator acknowledgment is required.[5]		0x20	F	Yes

[1] An offset of 0x10 or larger indicates an error requiring operator acknowledgment.

[2] This text may also be shown when the error in the *SPDU_ID* is due to an incorrect *SafetyBaseID*.

[3] This text may also be shown when the error in the *SPDU_ID* is due to an incorrect *SafetyStructureID*.

[4] This text may also be shown when the error in the *SPDU_ID* is due to an incorrect *SafetyProviderLevel*.

[5] A diagnostic message is generated only if the parameter *SPI.SafetyOperatorAckNecessary* is true, see transition T22 in Table 35.

*) The following classification is specified:

A) Transient communication error

B) Permanent communication error

C) Transmission quality seems not to be sufficient

D) Application error

E) Parameter error

F) Error does not affect communication itself.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

To avoid a flood of diagnostic messages in case of transmission errors, only up to two messages are shown even if multiple communication errors occur in sequence. This is ensured by the behaviour defined in the *SafetyConsumer*'s state machine.

Optional features (vendor-specific):

- Extend diagnostic data by expected value and received value, e.g.:
  Mismatch of *SafetyProviderID*:
  Expected *SafetyProviderID*: 0x00000005
  Received *SafetyProviderID*: 0x00000007
- Extend diagnostic data if a parameter of the *SafetyConsumer* is invalid.
  Example 1:
  The *SafetyConsumer* has been configured with invalid parameters.
  The value 0x00000000 is an invalid *SafetyProviderID*.

6.4.3 Method ReadSafetyDiagnostics of the SafetyProvider

This *Method* (as part of the *OPC UA mapper*) serves as a *Diagnostic Interface* and exists for each *SafetyProvider*. For time series observation, this interface can be polled, e.g. by a diagnostic device. For details, refer to the IEC 62541 *Information Model* described, see 6.2.2.4.

The *Diagnostic Interface Method* does not take any input parameters and returns both the input and output parameters of the last call of the *Method ReadSafetyData*.

Additionally, a 2-octet sequence number is added to the *Diagnostic Interface*, allowing for a detection of missed calls due to polling. The sequence number counts the number of accesses to *ReadSafetyData*.

A best practice recommendation is to store all input and output parameters if SComErr_diag is <> 0.

7 Safety communication layer protocol

7.1 General

This chapter describes in detail the protocol that is used for the *safety communication layer*.

7.2 SafetyProvider and SafetyConsumer

7.2.1 SPDU formats

7.2.1.1 General

Figure 11 shows the structure of a *RequestSPDU* which originates at the *SafetyConsumer* and contains a *SafetyConsumerID*, a *MonitoringNumber* (*MNR*), and one octet of (*non-safety-related*) *Flags*. See 7.2.1.2 to 7.2.1.4 for details. See 6.2.3.3 for details on the *RequestSPDUDataType* definition.

NOTE 1 The *RequestSPDU* does not contain a *CRC* signature.

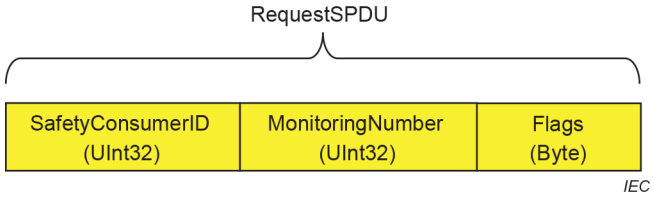


Figure 11 – RequestSPDU

Figure 12 shows the structure of a *ResponseSPDU* which originates at the *SafetyProvider* and contains the *SafetyData* (1 to 1 500 octets), an additional 25 octet safety code (*STrailer*) and the *NonSafetyData*. See 7.2.1.5 to 7.2.1.11 for details. See 6.2.3.4 for details on the *ResponseSPDUDataType* definition.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

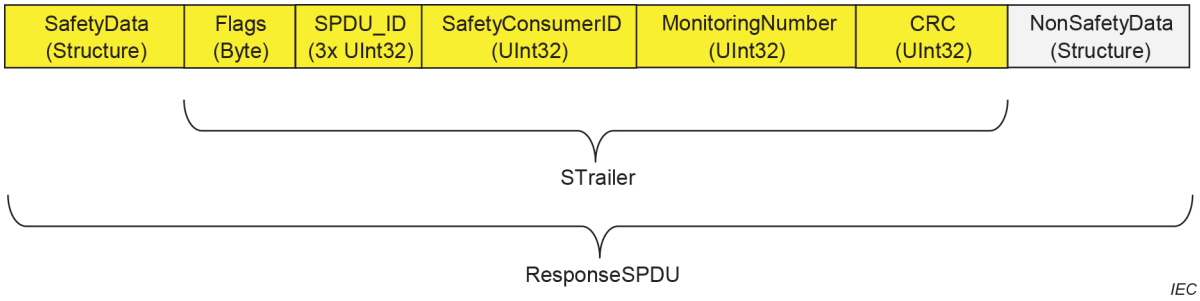


Figure 12 – ResponseSPDU

NOTE 2 To avoid spurious trips, the *ResponseSPDU* is transmitted in an atomic (consistent) way from the *OPC UA Platform Interface* of the *SafetyProvider* to the *OPC UA Platform Interface* of the *SafetyConsumer*. This is the task of the respective *OPC UA mapper*, see Figure 2.

7.2.1.2 RequestSPDU: SafetyConsumerID

Identifier of the *SafetyConsumer* instance, for diagnostic purposes, see 9.1.2.

7.2.1.3 RequestSPDU: MonitoringNumber

The *SafetyConsumer* uses the *MNR* to detect mis-timed *SPDUs*, e.g. such *SPDUs* which are continuously repeated by an erroneous network element which stores data. A different *MNR* is used in every *RequestSPDU* of a given *SafetyConsumer*, and a *ResponseSPDU* will only be accepted if its *MNR* matches the *MNR* of the corresponding *RequestSPDU*.

The checking for correctness of the *MNR* is only performed by the *SafetyConsumer*.

7.2.1.4 RequestSPDU: Flags

[RQ7.1] The *flags* of the *SafetyConsumer* (*RequestSPDU.Flags*) shall be used as shown in 6.2.3.1.

7.2.1.5 ResponseSPDU: SafetyData

[RQ7.2] *SafetyData* shall contain the safety-related application data transmitted from the *SafetyProvider* to the *SafetyConsumer*. It is comprised of a single or multiple basic IEC 62541 variables (see 6.2.5). For the sake of reducing distinctions of cases, *SafetyData* shall always be a structure, even if it comprised of only a single basic IEC 62541 *Variable*.

For the calculation of the *CRC* signature, the order in which this data is processed by the calculation is important. *SafetyProvider* and *SafetyConsumer* shall agree upon the number, type and order of application data transmitted in *SafetyData*. The sequence of *SafetyData* is fixed.

SafetyData may contain *qualifiers* for a fine-grained activation of *fail-safe* substitute values. For valid *process values*, the respective *qualifiers* are set to 1 (good), whereas the value 0 (bad) is used for invalid values. Invalid *process values* are replaced by *fail-safe substitute values* in the *SafetyConsumer*'s safety application. See 6.3.6.

7.2.1.6 ResponseSPDU: Flags

[RQ7.3] The *flags* of the *SafetyProvider* (*ResponseSPDU.Flags*) shall be used as shown in 6.2.3.2.

[RQ7.4] *Flags* in the *ResponseSPDU.Flags* which are reserved for future use shall be set to zero by the *SafetyProvider* and shall not be evaluated by the *SafetyConsumer*.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

7.2.1.7 ResponseSPDU: SPDU_ID

This field is used by the *SafetyConsumer* to check whether the *ResponseSPDU* is coming from the correct *SafetyProvider*. For details, see 7.2.3.1.

7.2.1.8 ResponseSPDU: SafetyConsumerID

[RQ7.5] The *SafetyConsumerID* in the *ResponseSPDU* shall be a copy of the *SafetyConsumerID* received in the corresponding *RequestSPDU*. See 7.2.3.1.

7.2.1.9 ResponseSPDU: MonitoringNumber

[RQ7.6] The *MonitoringNumber* in the *ResponseSPDU* shall be a copy of the *MonitoringNumber* received in the corresponding *RequestSPDU*. See 7.2.3.1.

NOTE The *SafetyConsumer* uses the *ResponseSPDU.MonitoringNumber* to detect *SPDUs* received with timeliness error, e.g. *SPDUs* which are continuously repeated by an erroneous network element which stores data. A different *MonitoringNumber* is used in every *RequestSPDU* of a given *SafetyConsumer*, and a *ResponseSPDU* will only be accepted if its *MonitoringNumber* matches the *MonitoringNumber* in the corresponding *RequestSPDU*.

7.2.1.10 ResponseSPDU: CRC

[RQ7.7] The *ResponseSPDU CRC* shall be used to detect data corruption. See 7.2.3.6 on how it is calculated in the *SafetyProvider* and how it is checked in the *SafetyConsumer*.

7.2.1.11 ResponseSPDU: NonSafetyData

[RQ7.8] This structure shall be used to transmit *NonSafetyData* values (e.g. diagnostic information) together with safe data consistently. *NonSafetyData* is not *CRC*-protected and can stem from an unsafe source.

[RQ7.9] When presented to the safety application (e.g. at an output of the *SafetyConsumer*), non-safety values shall clearly be indicated as "non-safety" by an appropriate vendor-specific mechanism (e.g. by using a different colour).

To avoid possible problems with empty structures, the dummy structure *NonSafetyDataPlaceholder* shall be used when no *NonSafetyData* is used (see requirement RQ6.8).

7.2.2 Behaviour

7.2.2.1 General

The two *SCL* services *SafetyProvider* and *SafetyConsumer* are specified using state diagrams.

7.2.2.2 SafetyProvider and SafetyConsumer Sequence diagram

Figure 13 and Figure 14 show sequences of requests and responses with *SafetyData* for this document using IEC 62541 *Client/Server* and *PubSub* communication mechanisms, respectively. The figures show selected scenarios only and are therefore informative.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

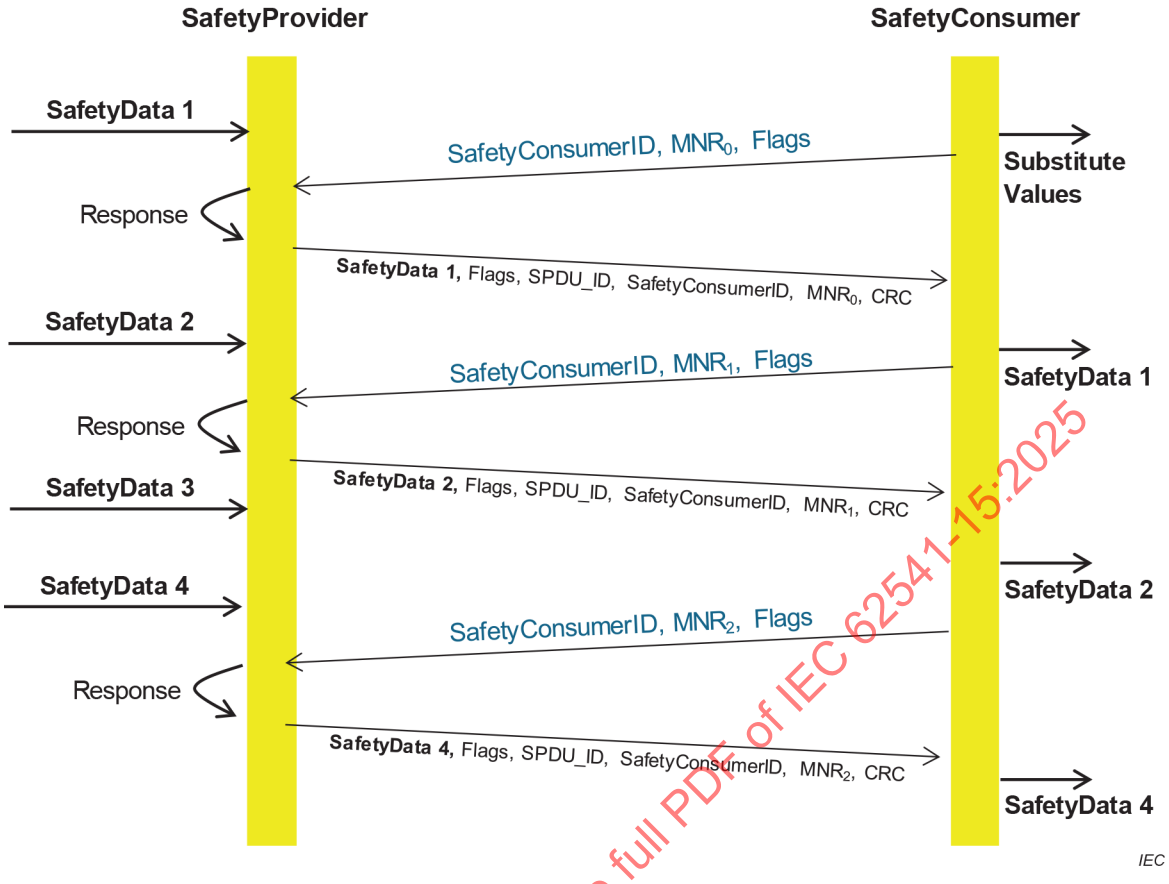


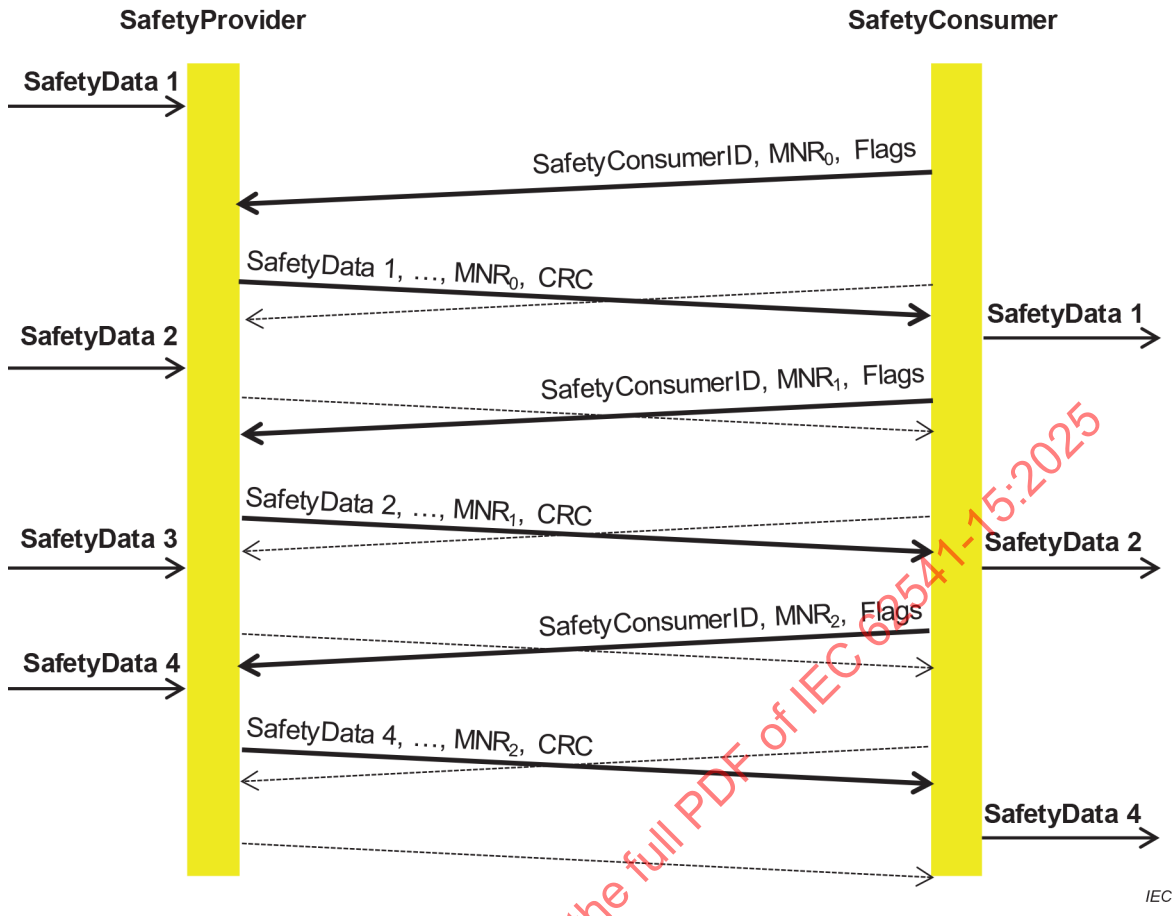
Figure 13 – Sequence diagram for requests and responses (Client/Server)

[RQ7.10] In the case of *Client/Server* communication, a *SafetyConsumer*'s *OPC UA mapper* may call a *SafetyProvider* (either the state machine implementation itself or the *SafetyProvider*'s *OPC UA mapper*) with an identical *RequestSPDU* multiple times in a row. In that case, the *SafetyProvider* (state machine or *OPC UA mapper*) shall answer all requests. The returned *ResponseSPDUs* may contain different values (e.g. currently available *process values*) or contain the initially returned values.

[RQ7.11] For each *SafetyProvider*, the implemented choice of behaviour according to RQ7.10 (i.e. whether currently available *process values* or initially returned values will be used) shall be documented in the corresponding safety manual.

NOTE 1 Since a *SafetyConsumer* checks for changed *MNRs* in *ResponseSPDUs* (see *SafetyConsumer* state S14_WaitForChangedSPDU in Table 34 and transition T17 in Table 35), and therefore will use only the first evaluated *ResponseSPDU* for a given *MNR* and all further *ResponseSPDUs* with the same *MNR* will be ignored.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025



NOTE The bold arrows represent communication with new data values, whereas dashed arrows contain repeated data values.

Figure 14 – Sequence diagram for requests and responses (PubSub)

NOTE 2 The state machines according to this document do not contain any retry-mechanisms to increase fault tolerance. In contrast, it is assumed that retry is already handled within the IEC 62541 stack (e.g. when using *Client/Server*, or by choosing a higher update rate for *PubSub*). The dashed lines therefore are not part of this document, but rather symbolize the repeated sending of data implemented in the IEC 62541 stack.

The *SafetyConsumerTimeout* is the watchdog time checked in the *SafetyConsumer*. The watchdog is restarted immediately before a new *RequestSPDU* is generated (transitions T14 and T28 of the *SafetyConsumer* in Table 35). If an appropriate *ResponseSPDU* is received in time, and the checks for data integrity, authenticity, and timeliness are all valid, the timer will not expire before it is restarted.

Otherwise, the watchdog timer expires, and the *SafetyConsumer* triggers a safe reaction. To duly check its timer, the *SafetyConsumer* is executed cyclically, with period *ConsumerCycleTime*. *ConsumerCycleTime* is expected to be smaller than *SafetyConsumerTimeout*.

The *ConsumerCycleTime* is the maximum time for the cyclic update of the *SafetyConsumer*. It is the timeframe from one execution of the *SafetyConsumer* to the next execution of the *SafetyConsumer*. The implementation of the monitoring of the *ConsumerCycleTime* and the reaction in case of exceeding the *ConsumerCycleTime* are not part of this document; these are vendor-specific.

[RQ7.12] The *ConsumerCycleTime* shall be monitored in a safety-related way.

[RQ7.26] A change of the *SafetyConsumerTimeout* parameter value shall take immediate effect on the *ConsumerTimer*.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

7.2.2.3 Duration of demand

In case it is necessary to ensure that a given *SafetyData* (e.g. a *safety* demand or a command value) that originates in the *SafetyProvider*'s safety application is being received by a *SafetyConsumer* and forwarded to the *SafetyConsumer*'s *safety* application, i.e. if no *SafetyData* in a series of *SafetyData* is to be missed, the following two cases shall be considered.

In case A, repeated identical *RequestSPDUs* are being answered by the *SafetyProvider* with *ResponseSPDUs* which contain the initially returned value. This is the case for *PubSub* communication and is a choice for *Client/Server* communication, see RQ7.11. In this case, the *SafetyProvider*'s safety application shall provide the respective *SafetyData* at the *SafetyProvider*'s *SAPI* until at least one change of the *MNR* is detected.

In case B, repeated identical *RequestSPDUs* are being answered by the *SafetyProvider* with the currently available *SafetyData*. This is a choice for *Client/Server* communication, see RQ7.11. In this case, the *SafetyProvider*'s *safety* application shall provide the respective *SafetyData* at the *SafetyProvider*'s *SAPI* until at least two changes of the *MNR* have been detected.

Figure 15 and Figure 16 show examples justifying case B by depicting two sequences of *ResponseSPDUs* as sent by a *SafetyProvider*. Due to the cycles of *SafetyProvider* and *SafetyConsumer* not being synchronized, a *SafetyConsumer* can evaluate any one of a given number of *ResponseSPDUs* for a given *RequestSPDU*.

In the examples, the *SafetyData* is made up of two components: the "respective *safety* data", i.e. a *safety* demand that is not to be missed (one of the values "A", "B" or "C") and non-demand numerical measurement values for which it does not matter whether some are not received by the *SafetyConsumer*.

Since *NonSafetyData* is consistently transmitted with *SafetyData*, the same considerations apply for *NonSafetyData*.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

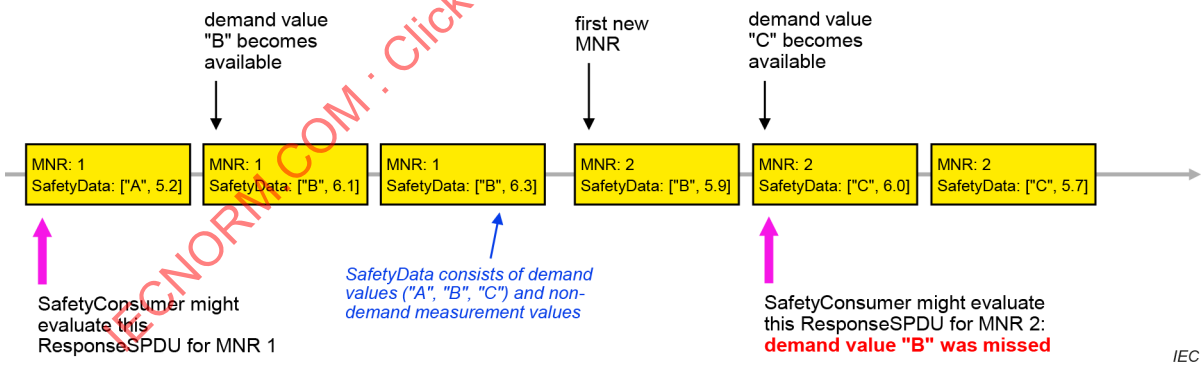


Figure 15 – Duration of demand example for missed demand value in case of currently available SafetyData not being provided until second change of MNR

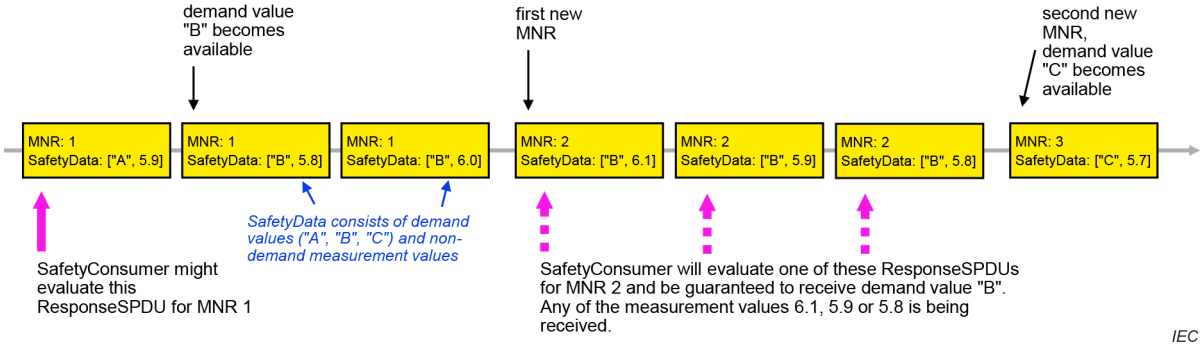


Figure 16 – Duration of demand example for received demand value in case of currently available SafetyData being provided

7.2.2.4 SafetyProvider state diagram

[RQ7.13] Figure 17 shows a simplified representation of the state diagram of the *SafetyProvider*. The exact behaviour is described in Table 30, Table 31, and Table 32. The *SafetyProvider* shall implement that behaviour. It is not required to literally follow the entries given in the tables, if the externally observable behaviour does not change.

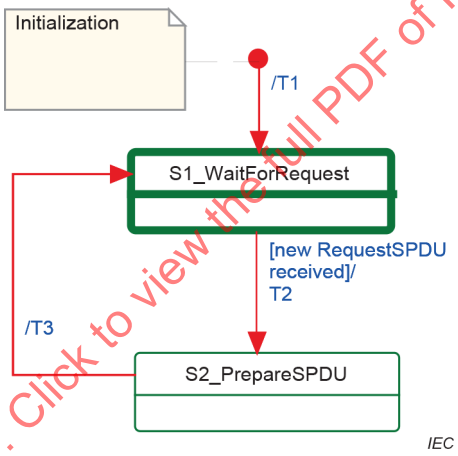
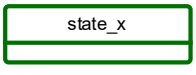
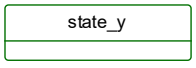


Figure 17 – Simplified representation of the state diagram for the SafetyProvider

Table 29 shows the symbols used for state machines.

Table 29 – Symbols used for state machines

Graphical representation	Type	Description
	Activity state	Within these interruptible activity states the *SafetyProvider* waits for new input.
	Action state	Within these non-interruptible action states events like new requests are deferred until the next activity state is reached, see [1][1].

[1] Numbers in square brackets refer to the Bibliography.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

The transitions are fired in case of an event. "Event" in this context means either a *Method* call in the case of *Client/Server* communication or the detection of a changed *RequestSPDU* by the *OPC UA mapper* in the case of *PubSub* communication.

In case of several possible transitions, so-called guard conditions (refer to […] in UML diagrams) define which transition to fire.

The diagram consists of activity and action states. Activity states are surrounded by bold lines, action states are surrounded by thin lines. While activity states can be interruptible by new events, action states are not. External events occurring while the state machine is in an action state, are deferred until the next activity state is reached.

NOTE The details on how to implement activity states and action states are vendor-specific. Typically, in a real-time system the task performing the *SafetyProvider* or *SafetyConsumer* state machine is executed cyclically (see 6.3.5). Whenever the task is woken up by the scheduler of the operating system while it is in an action state, it executes action states until its time slice is used up, or an activity state is reached. Whenever a task being in an activity state is woken up, it checks for input. If no new input is available, it immediately returns to the sleep state without changing state.

If input is available, it starts executing action states until its time-slice is up or until the next activity state is reached.

Table 30 shows the internal items of a *SafetyProvider* instance.

Table 30 – SafetyProvider instance internal items

Internal items	Type	Definition
RequestSPDU_i	Variable	Local memory for *RequestSPDU* (required to react on changes).
SPDU_ID_1_i	UInt32	Local variable to store *SPDU_ID_1*.
SPDU_ID_2_i	UInt32	Local variable to store *SPDU_ID_2*.
SPDU_ID_3_i	UInt32	Local variable to store *SPDU_ID_3*.
BaseID_i	Guid	Local variable containing the BaseID (taken either from the *SPI* or *SAPI*).
ProviderID_i	UInt32	Local variable containing the ProviderID (taken either from the *SPI* or *SAPI*).
<Get RequestSPDU>	Macro	Instruction to take the whole *RequestSPDU* from the *OPC UA mapper*.
<Set ResponseSPDU>	Macro	Instruction to transfer the whole *ResponseSPDU* to the *OPC UA mapper*.

<Calc SPDU_ID_i>	Macro	const uint32 SafetyProviderLevel_ID:= … // see 7.2.3.3
		
		If(SAPI.SafetyBaseID == 0) then
		BaseID_i:= SPI.SafetyBaseIDConfigured
		Else
		BaseID_i:= SAPI.SafetyBaseID
		
		Endif
		If(SAPI.SafetyProviderID == 0) then
		ProviderID_i:= SPI.SafetyProviderIDConfigured
		Else
		ProviderID_i:= SAPI.SafetyProviderID
		
		Endif
		
		SPDU_ID_1_i:= BaseID_i (octets 0…3)
		XOR SafetyProviderLevel_ID
		
		SPDU_ID_2_i:= BaseID_i (octets 4…7)
		XOR SPI.SafetyStructureSignature
		
		SPDU_ID_3_i:= BaseID_i (octets 8…11)
		XOR BaseID_i (octets 12…15)
		XOR ProviderID_i
		
		// see 7.2.3.2 for clarification

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Internal items	Type	Definition
<build ResponseSPDU>	Macro	Take the *MNR* and the *SafetyConsumerID* of the received *RequestSPDU*. Add the SPDU_ID_1_i, SPDU_ID_2_i, SPDU_ID_3_i, Flags, the *SafetyData* and the *NonSafetyData*, as well as the calculated *CRC*. See 7.2.3.1

Table 31 shows the states of a *SafetyProvider* instance. The *SafetyProvider* does not check for correct configuration. It will reply to requests even if it is incorrectly configured (e.g. its *SafetyProviderID* is zero). However, *SafetyConsumers* will never try to communicate with *SafetyProviders* having incorrect parameters, see Transitions T13 and T27 in Table 35 and the macro <ParametersOK?> in Table 33.

Table 31 – States of SafetyProvider instance

State name	State description
Initialization	// Initial state SAPI.SafetyData:= 0 SAPI.NonSafetyData:= 0 SAPI.MonitoringNumber:= 0 SAPI.SafetyConsumerID:= 0 SAPI.OperatorAckRequested:= 0 RequestSPDU_i:= 0
S1_WaitForRequest	// waiting on next *RequestSPDU* from *SafetyConsumer* <Get RequestSPDU>
S2_PrepareSPDU	ResponseSPDU.Flags.ActivateFSV:= SAPI.ActivateFSV ResponseSPDU.Flags.OperatorAckProvider:= SAPI.OperatorAckProvider ResponseSPDU.Flags.TestModeActivated:= SAPI.EnableTestMode <Calc SPDU_ID_i> <build ResponseSPDU> // see 7.2.3.1

Table 32 shows the transitions of the *SafetyProvider*.

Table 32 – SafetyProvider transitions

Transition	Source state	Target state	Guard condition	Activity
T1	Init	S1		
T2	S1	S2	// **RequestSPDU received**[1] -	// **Process request** RequestSPDU_i:= RequestSPDU SAPI.MonitoringNumber:= RequestSPDU.MonitoringNumber SAPI.SafetyConsumerID:= RequestSPDU.SafetyConsumerID SAPI.OperatorAckRequested:= RequestSPDU.Flags.OperatorAckRequested
T3	S2	S1	// **SPDU is prepared** -	<Set ResponseSPDU>

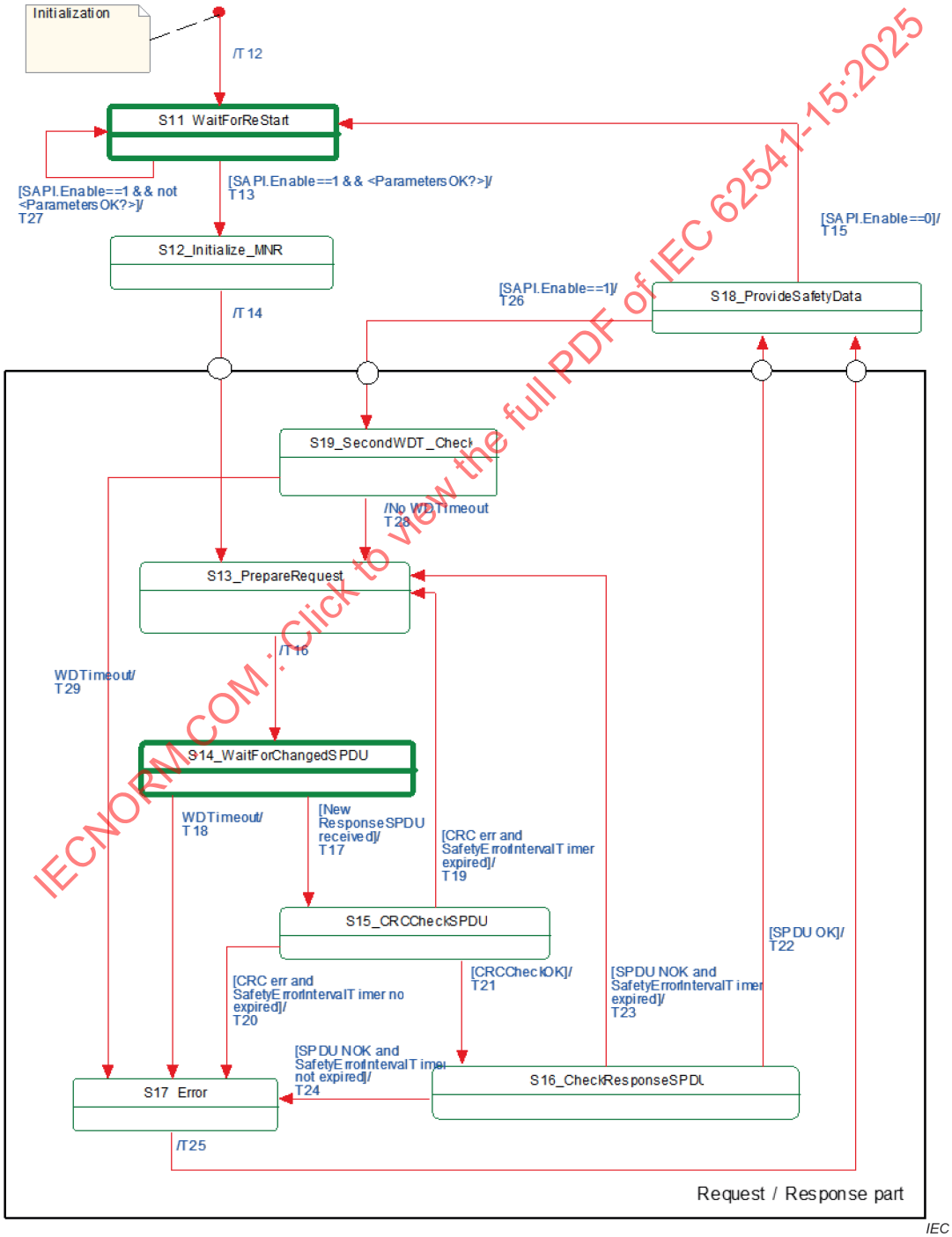
[1] See the preceding explanation in 7.2.2.3 of what constitutes events which trigger this transition.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

7.2.2.5 SafetyConsumer state diagram

[RQ7.14] Figure 18 shows a simplified representation of the state diagram of the *SafetyConsumer*. The exact behaviour is described in Table 33, Table 34, and Table 35. The *SafetyConsumer* shall implement this behaviour. It is not required to literally follow the entries given in the tables, if the externally observable behaviour does not change.

To avoid unnecessary spurious trips requiring operator acknowledgment, it is recommended that a *SafetyConsumer* is started after an IEC 62541 connection to a running *SafetyProvider* has been established, or that the setting of input *SAPI.Enable* to "1" is delayed until the *SafetyProvider* is running.



IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Figure 18 – Principle state diagram for SafetyConsumer

Table 33 shows the internal items of a *SafetyConsumer*. A macro is a shorthand representation for operations described in the according definition.

Table 33 – SafetyConsumer internal items

Internal items	Type	Definition
Constants		
MNR_min:= 0x100	UInt32	// 0x100 is the start value for *MNR*, also used after wrap-around // The values 0...0xFF are reserved for future use
Variables		
FaultReqOA_i	Boolean	Local memory for errors which request operator acknowledgment
OperatorAckConsumerAllowed_i	Boolean	Auxiliary *flag* indicating that operator acknowledgment is allowed. It is true if the input *SAPI.OperatorAckConsumer* has been 'false' since FaultReqOA_i was set
MNR_i	UInt32	Local *MonitoringNumber* (*MNR*)
prevMNR_i	UInt32	Local memory for previous *MNR*
ConsumerID_i	UInt32	Local memory for *SafetyConsumerID* in use
CRCCheck_i	Boolean	Local variable used to store the result of the *CRC* check
SPDUCheck_i	Boolean	Local variable used to store the result of the additional *SPDU* checks
SPDU_ID_1_i	UInt32	Local variable to store the expected *SPDU_ID_1*
SPDU_ID_2_i	UInt32	Local variable to store the expected *SPDU_ID_2*
SPDU_ID_3_i	UInt32	Local variable to store the expected *SPDU_ID_3*
SPI_SafetyConsumerID_i	UInt32	Local variable to store the parameter SafetyConsumerID
SPI_SafetyProviderID_i	UInt32	Local variable to store the parameter *SafetyProviderID*
SPI_SafetyBaseID_i	UInt128	Local variable to store the parameter *SafetyBaseID*
SPI_SafetyStructureSignature_i	UInt32	Local variable to store the parameter *SafetyStructureSignature*
SPI_SafetyOperatorAckNecessary_i	Boolean	Local variable to store the parameter SafetyOperatorAckNecessary
SPI_SafetyErrorIntervalLimit_i	UInt16	Local variable to store the parameter *SafetyErrorIntervalLimit*
MNR_re_sync_i	Boolean	Local variable used to support re-synchronization of *MNR* after detected error
Timers		
ConsumerTimer	Timer	This timer is used to check whether the next valid *ResponseSPDU* has arrived on time. It is initialized using the parameter *SPI.SafetyConsumerTimeout*. NOTE 1 As opposed to other parameters, a modification of the parameter value *SafetyConsumerTimeout* takes effect immediately, i.e. not only when state S11 is visited, see RQ7.26.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Internal items	Type	Definition
ErrorIntervalTimer	Timer	This timer is used to check the elapsed time between errors. If the elapsed time between two consecutive errors is smaller than the value *SafetyErrorIntervalLimit*, *FSV* will be activated. Otherwise, the *ResponseSPDU* is discarded and the *SafetyConsumer* waits for the next *ResponseSPDU*. This timer is initialized using the local variable SPI_SafetyErrorIntervalLimit_i. See Table 26, 6.3.4.3, and 9.4 for more information. NOTE 2 The local variable SPI_SafetyErrorIntervalLimit_i is not to be confused with the parameter *SPI.SafetyErrorIntervalLimit*. The local variable is copied from the parameter in state S11 (restart). Hence, if the parameter value changes during runtime, the new value will only be used after the connection has been restarted.
Macros <...><...>		

<risingEdge x>	Macro	// detection of a rising edge:
		
		If x==true && tmp==false
		Then
		result:= true
		Else
		result:= false
		Endif
		
		tmp:= x

<Get ResponseSPDU>	Macro	Instruction to take the whole *ResponseSPDU* from the *OPC UA mapper*.

<Use FSV>	Macro	SAPI.SafetyData is set to binary 0
		
		If [<ConsumerTimer expired || SAPI.Enable==0 ?>]
		Then
		SAPI.NonSafetyData is set to binary 0
		
		Else
		SAPI.NonSafetyData is set to ResponseSPDU.NonSafetyData
		
		Endif
		
		SAPI.FSV_Activated:= 1
		
		RequestSPDU.Flags.FSV_Activated:= 1
		
		NOTE 3 If a safety application prefers *fail-safe* values other than binary 0, this can be implemented in the safety application by querying SAPI.FSV_Activated.
		
		NOTE 4 The *NonSafetyData* is always updated if data is available. In case of a timeout, no data is available, which is indicated using binary zero. If it is necessary for an application to distinguish between "no data available" and "binary zero received", it can add a Boolean variable to the *NonSafetyData*. This value is set to "1" during normal operation, and to "0" for indicating that no data is available.

<Use PV>	Macro	SAPI.SafetyData is set to ResponseSPDU.SafetyData SAPI.NonSafetyData is set to ResponseSPDU.NonSafetyData SAPI.FSV_Activated:= 0 RequestSPDU.Flags.FSV_Activated:= 0 RequestSPDU.Flags.CommunicationError:= 0
<Set RequestSPDU>	Macro	Instruction to transfer the whole *RequestSPDU* to the *OPC UA mapper*.
<(Re)Start ConsumerTimer>	Macro	Restarts the *ConsumerTimer*.
<(Re)Start ErrorIntervalTimer>	Macro	Restarts the *ErrorIntervalTimer*.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Internal items	Type	Definition
<ConsumerTimer expired?>	Macro	Yields "true" if the timer is running longer than SPI.SafetyConsumerTimeout since last restart, "false" otherwise.
<ErrorIntervalTimer expired?>	Macro	Yields "true" if the timer is running longer than SPI.SafetyErrorIntervalLimit since last restart, "false" otherwise.

<Assign ConsumerID>	Macro	**If** SAPI.SafetyConsumerID != 0
		Then
		ConsumerID_i:= SAPI.SafetyConsumerID
		Else
		ConsumerID_i:= SPI_SafetyConsumerID_i
		Endif
		RequestSPDU.SafetyConsumerID:= ConsumerID_i

<Calc SPDU_ID_i>	Macro	uint128 BaseID
		uint32 ProviderID
		const uint32 SafetyProviderLevel_ID:= ... // see 7.2.3.3
		If(SAPI.SafetyBaseID == 0)
		Then
		BaseID:= SPI_SafetyBaseID_i
		Else
		BaseID:= SAPI.SafetyBaseID
		Endif
		If(SAPI.SafetyProviderID == 0)
		Then
		ProviderID:= SPI_SafetyProviderID_i
		Else
		ProviderID:= SAPI.SafetyProviderID
		Endif
		SPDU_ID_1_i:= BaseID (octets 0…3)
		XOR SafetyProviderLevel_ID
		SPDU_ID_2_i:= BaseID (octets 4…7)
		XOR SPI_SafetyStructureSignature_i
		SPDU_ID_3_i:= BaseID (octets 8…11)
		XOR BaseID (octets 12…15)
		XOR ProviderID
		// see 7.2.3.2 for clarification

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Internal items	Type	Definition

<ParametersOK?>	Macro	Boolean result:= true
		If(SAPI.SafetyBaseID == 0 && SPI_SafetyBaseID_i==0)
		Then
		result:= false
		Else
		Endif
		
		If(SAPI.SafetyProviderID == 0 && SPI_SafetyProviderID_i==0)
		Then
		result:= false
		Else
		Endif
		
		If(SAPI.SafetyConsumerID == 0 && SPI_SafetyConsumerID_i==0)
		Then
		result:= false
		Else
		Endif
		
		If(SPI_SafetyStructureSignature_i==0)
		Then
		result:= false
		Else
		Endif
		
		yield result

<Set Diag(ID,
Boolean isPermanent)>
	Macro	// ID is the identifier for the type of diagnostic output, see Table 28.
		// Parameter *isPermanent* is used to indicate a permanent error.
		// Only one diagnostic message is created for multiple permanent
		// errors in sequence
		
		If(RequestSPDU.Flags.CommunicationError == 0)
		Then
		<do vendor-specific function for diagnostic output using ID>
		Else
		//do nothing
		Endif
		
		RequestSPDU.Flags.CommunicationError:= isPermanent
		//
		
		NOTE 5 See Table 28 for possible values for "ID" and their codes.

<ResponseSPDU ready for checks>	Macro	Boolean result:= false
		
		If MNR_re_sync_i == false
		
		Then
		If ResponseSPDU.MNR <> prevMNR_i
		
		Then result:= true
		
		Else
		//do nothing
		Endif
		
		Else
		
		If ResponseSPDU.MNR == MNR_i
		
		Then result:= true
		
		Else
		//do nothing
		Endif
		
		Endif
		
		yield result

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Internal items	Type	Definition

<Handle WDTimeout>	Macro	<Set Diag(CommErrTO,isPermanent:=true)>
		<Use FSV>
		If SPI_SafetyOperatorAckNecessary_i == 1
		Then
		FaultReqOA_i:= 1
		SAPI.OperatorAckRequested:= 0
		RequestSPDU.Flags.OperatorAckRequested:= 0
		Else
		// do nothing
		Endif

External event		
Restart cycle	Event	The external call of *SafetyConsumer* can be interpreted as event "restart cycle"

Table 34 shows the states of the *SafetyConsumer*. The *SafetyConsumer* parameters are accessed only in state S11. In this state, a copy is made, and in all other states and transitions the copied values are used. This ensures that a change of one of these parameters takes effect only when a new safety connection is established. The only exception from this rule is the parameter *SafetyConsumerTimeout*. A change of this parameter becomes effective immediately (see RQ7.26). If this is not the desired behaviour, i.e. if parameters should be changeable during runtime, this can be accomplished by establishing a second safety connection according to this document with the new parameters, and then switching between these two safety connections at runtime.

Table 34 – SafetyConsumer states

State name	State description
Initialization	// Initial state of the *SafetyConsumer* instance. <Use FSV> SAPI.OperatorAckRequested:= 0 RequestSPDU.Flags.OperatorAckRequested:= 0 SAPI.OperatorAckProvider:= 0 FaultReqOA_i:= 0 OperatorAckConsumerAllowed_i:= 0 SAPI.TestModeActivated:= 0 RequestSPDU.Flags.CommunicationError:= 0 MNR_re_sync_i:= false

S11_Wait for (Re)Start	// Safety layer is waiting (Re)Start
	// Changes to these parameters are only considered in this state
	// Exception: a change of *SafetyConsumerTimeout* is possible during operation
	// Read parameters from the *SPI* and store them in local variables:
	SPI_SafetyConsumerID_i:= SPI.SafetyConsumerID
	SPI_SafetyProviderID_i:= SPI.SafetyProviderIDConfigured
	SPI_SafetyBaseID_i:= SPI.SafetyBaseIDConfigured
	SPI_SafetyStructureSignature_i:= SPI.SafetyStructureSignature
	SPI_SafetyOperatorAckNecessary_i:= SPI.SafetyOperatorAckNecessary
	SPI_SafetyErrorIntervalLimit_i:= SPI_SafetyErrorIntervalLimit

S12_initialize MNR	// Use previous *MNR* if known
	// or random *MNR* within the allowed range (e.g. after cold start), see 9.2.
	MNR_i:= (previous MNR_i if known) or (random *MNR*)
	MNR_i:= max(MNR_i, MNR_min)[1]

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

State name	State description
S13_PrepareRequest	// Build *RequestSPDU* and send (done in T16)
S14_WaitForChangedSPDU	// Safety Layer is waiting for next *ResponseSPDU* from *SafetyProvider*. // A new *ResponseSPDU* is characterized by a change in the *MNR*.

S15_CRCCheckSPDU	// Check *CRC*
	uint32 CRC_calc
	CRCCheck_i:= (CRC_calc == ResponseSPDU.CRC)
	// see 7.2.3.6 on how to calculate CRC_calc

S16_CheckResponseSPDU	// Check *SafetyConsumerID* and *SPDU_ID* and *MNR* (see T22, T23, T24) SPDUCheck_i:= ResponseSPDU.SPDU_ID_1 == SPDU_ID_1_i && ResponseSPDU.SPDU_ID_2 == SPDU_ID_2_i && ResponseSPDU.SPDU_ID_3 == SPDU_ID_3_i && ResponseSPDU.SafetyConsumerID == ConsumerID_i && ResponseSPDU.MNR == MNR_i
S17_Error	SAPI.TestModeActivated:= 0
S18_ProvideSafetyData	// Provide *SafetyData to the application program*
S19_SecondWDT_Check	// Second check of WDTimeout // Prevents restarting of *ConsumerTimer* if it expired since initial check
[1] This ensures that the *MNR* is greater or equal to MNR_min, in cases the random number generator yielded a smaller value.	

Table 35 shows the transitions of the *SafetyConsumer*.

Table 35 – SafetyConsumer transitions

Transition	Source state	Target state	Guard condition	Activity
T12	Init	S11	-	
T13	S11	S12	**//Start** [SAPI.Enable==1 && <ParametersOK?>]	<(Re)Start ErrorIntervalTimer> <calc SPDU_ID_i> // see 7.2.3.2 for clarification
T14	S12	S13	**// *MNR* initialized**	<(Re)Start ConsumerTimer> <Assign ConsumerID>

T15	S13	S11	**// Termination**
			[SAPI.Enable==0]
				<Use FSV>
				RequestSPDU.Flags.CommunicationError:= 0
				// necessary to make sure that no diagnostic
				// message is lost, see macro <Set Diag ...>
				// NOTE Depending on its implementation, it could
				// be necessary to stop the *ConsumerTimer* here.

T16	S13	S14	**// Build RequestSPDU**
			// and send it
				prevMNR_i:= MNR_i
				If MNR_i == 0xFFFFFFFFF
				Then
				MNR_i:= MNR_min
				Else
				MNR_i:= MNR_i + 1
				Endif
				RequestSPDU.MonitoringNumber:= MNR_i
				<Set RequestSPDU>

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Transition	Source state	Target state	Guard condition	Activity
T17	S14	S15	**// Changed ResponseSPDU // is received1** <Get ResponseSPDU>[2] <ResponseSPDU ready for checks>	// A changed *ResponseSPDU* is characterized by a change in the *MNR*.
T18	S14	S17	**// WDTimeout** [<ConsumerTimer expired?>]	<Handle WDTimeout>
T19	S15	S13	**// When CRC err and ErrorIntervalTimer expired** [(crcCheck_i == 0) && <ErrorIntervalTimer expired?>]	MNR_re_sync_i:= true <(Re)Start ErrorIntervalTimer> <Set Diag(CRCerrIgn, isPermanent:=false)>
T20	S15	S17	**// When CRC err and ErrorIntervalTimer not expired** [(crcCheck_i == 0) && not <ErrorIntervalTimer expired?>]	<(Re)Start ErrorIntervalTimer> <Set Diag(CRCerrOA, isPermanent:=true)> <Use FSV> FaultReqOA_i:=1 SAPI.OperatorAckRequested:= 0 RequestSPDU.Flags.OperatorAckRequested:= 0
T21	S15	S16	**// When CRCCheckOK** [crcCheck_i == 1]	-

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Transition	Source state	Target state	Guard condition	Activity

T22	S16	S18	// **SPDU OK**
			[SPDUCheck_i==true]
				// For clarification, refer to Figure 19;
				MNR_re_sync_i:= false
				// indicate OA from provider
				SAPI.OperatorAckProvider:= ResponseSPDU.Flags.OperatorAckProvider
				// OA requested due to rising edge at ActivateFSV?
				If (<risingEdge ResponseSPDU.Flags.ActivateFSV>&& SPI_SafetyOperatorAckNecessary_i == true)
				Then
				FaultReqOA_i:= 1;
				<Set Diag(FSV_Requested,isPermanent:=true)>
				Else
				// do nothing
				Endif
				// *Set Flags if OA requested:*
				If FaultReqOA_i==1
				Then
				SAPI.OperatorAckRequested:= 1,
				RequestSPDU.Flags.OperatorAckRequested:= 1,
				OperatorAckConsumerAllowed_i:= 0,
				FaultReqOA_i:= 0
				Else
				//do nothing
				Endif
				// Wait until OperatorAckConsumer is not active
				If SAPI.OperatorAckConsumer==0
				Then
				OperatorAckConsumerAllowed_i:= 1
				Else
				//do nothing
				Endif
				// *Reset Flags after OA:*
				If SAPI.OperatorAckConsumer ==1 && OperatorAckConsumerAllowed_i == 1
				Then
				SAPI.OperatorAckRequested:= 0,
				RequestSPDU.Flags.OperatorAckRequested:= 0
				Else
				// do nothing
				Endif
				If SAPI.OperatorAckRequested==1 || ResponseSPDU.Flags.ActivateFSV==1
				Then <Use FSV>
				Else <Use PV>
				Endif
				// Notify safety application that *SafetyProvider* is in test mode:
				SAPI.TestModeActivated:= ResponseSPDU.Flags.TestModeActivated

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Transition	Source state	Target state	Guard condition	Activity

T23	S16	S13	// **SPDU NOK and ErrorIntervalTimer expired**
			
			[SPDUCheck_i == false && <ErrorIntervalTimer expired?>]
				<(Re)Start ErrorIntervalTimer>,
				
				MNR_re_sync_i:= true
				
				// Send diagnostic message according to the
				// detected error:
				
				If ResponseSPDU.SafetyConsumerID <> ConsumerID_i
				Then <Set Diag(CoIDerrIgn, isPermanent:=false)>
				Else
				If ResponseSPDU.MNR<>MNR_i
				Then <Set Diag(MNRerrIgn, isPermanent:=false)>
				Else
				//do nothing
				EndIf
				If ResponseSPDU.SPDU_ID_1<> SPDU_ID_1_i ||
				ResponseSPDU.SPDU_ID_2<> SPDU_ID_2_i ||
				ResponseSPDU.SPDU_ID_3<> SPDU_ID_3_i
				Then
				<Set Diag(SD_IDerrIgn,
				isPermanent:=false)>[3]
				Else
				// do nothing
				Endif
				Endif

T24	S16	S17	// **SPDU NOK and ErrorIntervalTimer not expired**
			
			[SPDUCheck_i == 0 && not <ErrorIntervalTimer expired?>]
				<(Re)Start ErrorIntervalTimer>
				// Send diagnostic message according to the
				// detected error:
				
				If ResponseSPDU.SafetyConsumerID<> ConsumerID_i
				Then <Set Diag(CoIDerrOA, isPermanent:=true)>
				Else
				If ResponseSPDU.MNR<>MNR_i
				Then
				<Set Diag(MNRerrOA,isPermanent:=true)>
				Else
				//do nothing
				Endif
				If ResponseSPDU.SPDU_ID_1<> SPDU_ID_1_i ||
				ResponseSPDU.SPDU_ID_2<> SPDU_ID_2_i ||
				ResponseSPDU.SPDU_ID_3<> SPDU_ID_3_i
				Then <Set Diag(SD_IderrOA,isPermanent:=true)>
				Else
				//do nothing
				Endif
				Endif
				
				FaultReqOA_i:= 1
				
				SAPI.OperatorAckRequested:= 0
				RequestSPDU.Flags.OperatorAckRequested:= 0
				<Use FSV>

T25	S17	S18	// **SPDU NOK** -	MNR_re_sync_i:= true
T26	S18	S19	// **Restart Cycle** [SAPI.Enable==1]	-

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Transition	Source state	Target state	Guard condition	Activity
T27	S11	S11	**// Invalid parameters** [SAPI.Enable==1 && not <ParametersOK?>]	<Set Diag(ParametersInvalid, isPermanent:=true)>
T28	S19	S13	**// No WDTimeout**	<(Re)Start ConsumerTimer>
T29	S19	S17	**// WDTimeout** [<ConsumerTimer expired?>]	<Handle WDTimeout>

[1] Another event like "*Method* completion successful" can be used as guard condition of "Changed *ResponseSPDU* received" as well.

[2] *SPDUs* with all values (incl. *CRC* signature) being zero shall be ignored, see requirement RQ5.6.

[3] See Table 28.

7.2.2.6 SafetyConsumer sequence diagram for operator acknowledgment (informative)

Figure 19 shows the sequence after the detection of an error requiring operator acknowledge until communication returns to delivering process values again.

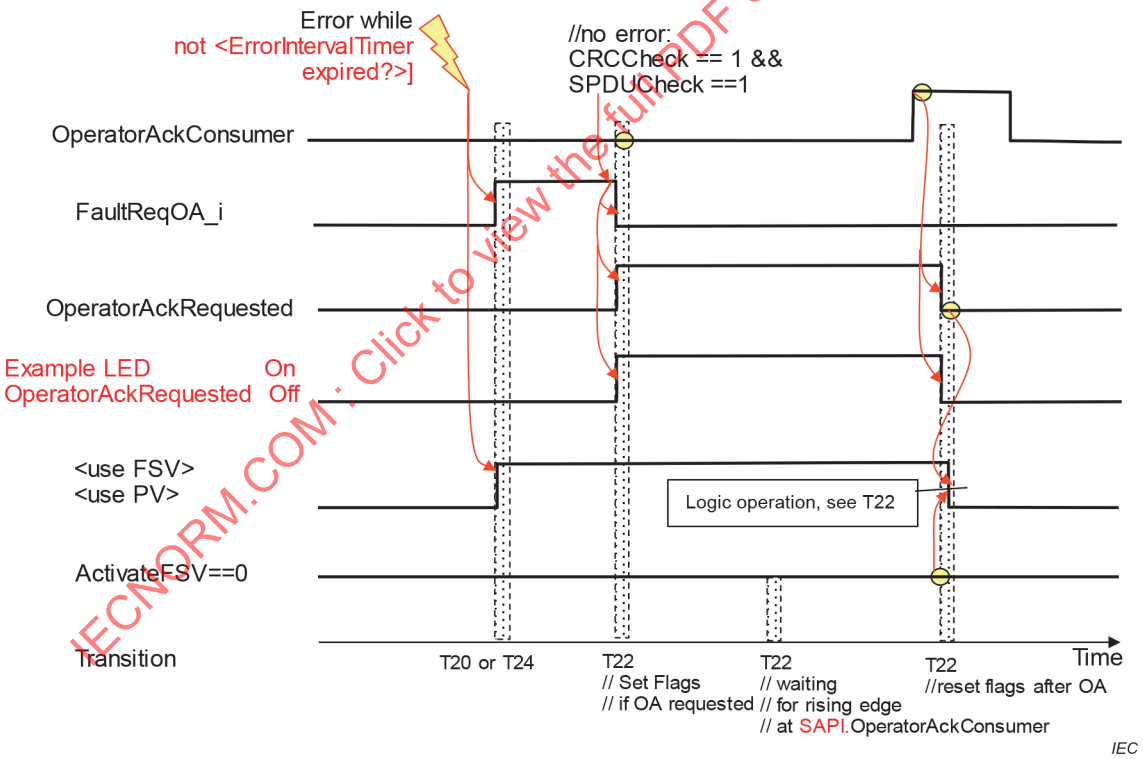


Figure 19 – Sequence diagram for OA

After the error is gone the sequence follows the logic of T22 in Table 35.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

7.2.3 Subroutines

7.2.3.1 Build ResponseSPDU

[RQ7.15] The *ResponseSPDU* shall be built by the *SafetyProvider* by copying *RequestSPDU.MonitoringNumber* and *RequestSPDU.SafetyConsumerID* into the *ResponseSPDU*. In addition, *SPDU_ID*, *Flags*, the *SafetyData* and the *NonSafetyData* shall be updated. Finally, *ResponseSPDU.CRC* shall be calculated and appended.

Figure 20 gives an overview over the task of building the *ResponseSPDU*.

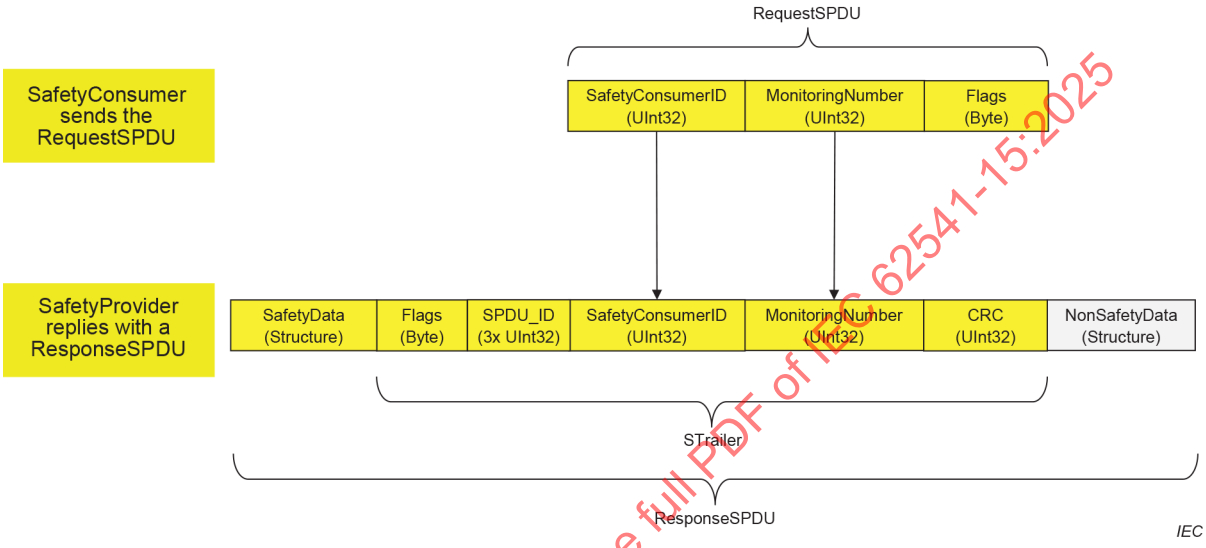


Figure 20 – Overview of task for SafetyProvider

For the *ResponseSPDU.Flags*, see 7.2.1.6. For the calculation of the *SPDU_ID*, see 7.2.3.2. For the calculation of the *CRC*, see 7.2.3.6.

7.2.3.2 Calculation of the SPDU_ID_1, SPDU_ID_2, SPDU_ID_3

[RQ7.16] *SPDU_ID_1*, *SPDU_ID_2* and *SPDU_ID_3* shall be calculated according to Figure 21 and Table 36.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

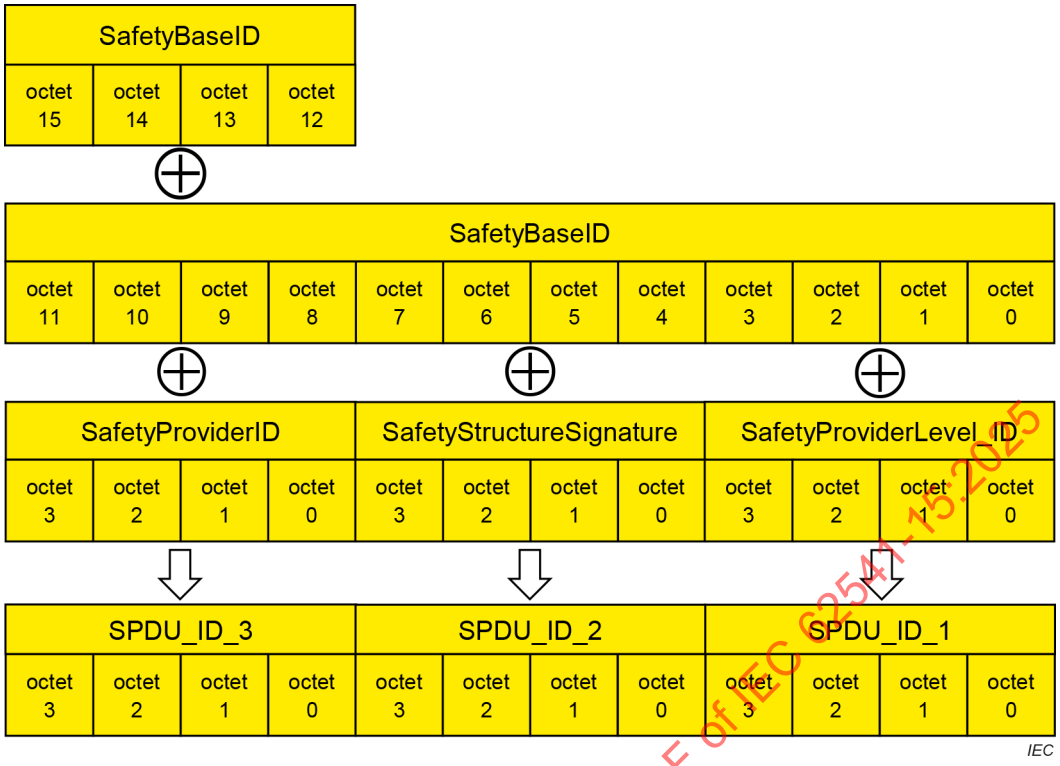


Figure 21 – Calculation of the SPDU_ID

Table 36 – Presentation of the SPDU_ID

SPDU_ID_1:= SafetyBaseID (octets 0…3) **XOR** SafetyProviderLevel_ID (octets 0…3)
SPDU_ID_2:= SafetyBaseID (octets 4…7) **XOR** SafetyStructureSignature (octets 0…3)
SPDU_ID_3:= SafetyBaseID (octets 8…11) **XOR** SafetyBaseID (octets 12…15) **XOR** SafetyProviderID (octets 0…3)

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

In case of a mismatch between expected *SPDU_ID* and actual *SPDU_ID*, the following rules can be used for diagnostic purposes:

- If all of *SPDU_ID_1*, *SPDU_ID_2*, and *SPDU_ID_3* differ, there probably is a mismatching *SafetyBaseID*.
- If *SPDU_ID_3* differs, but *SPDU_ID_1* and *SPDU_ID_2* do not, there is a mismatching *SafetyProviderID*.
- If *SPDU_ID_2* differs, but *SPDU_ID_1* and *SPDU_ID_3* do not, the structure or identifier of the *SafetyData* do not match.
- If *SPDU_ID_1* differs, but *SPDU_ID_2* and *SPDU_ID_3* do not, the *SafetyProviderLevel* does not match.

By these rules, there is a very low probability ($<10^{-9}$) that a mismatching *SafetyBaseID* will be misinterpreted. From a practical view, this probability can be ignored.

7.2.3.3 Example for the calculation of SPDU_ID_1, SPDU_ID_2 and SPDU_ID_3 (informative)

Figure 22 shows a concrete example of the calculation of *SPDU_ID_1*, *SPDU_ID_2* and *SPDU_ID_3*. The following input values were chosen for the calculation: *SafetyBaseID* is the *GUID* "72962B91-FA75-4AE6-8D28-B404DC7DAF63", *SafetyProviderID* has the value 0xE0EA6B40, *SafetyStructureSignature* has the value 0xDE7329FD and the *SafetyProviderLevel_ID* is chosen as 0xDEAA9DEE (representing *SIL*3).

See IEC 62541-6:2020, 5.2.2.6 for details on the encoding of *Guids*. The octets from the resulting octet stream are used according to Figure 21, i.e. in "reverse order".

The resulting *SPDU_IDs* are as follows: *SPDU_ID_1* has the value of 0xAC3CB67F, *SPDU_ID_2* has the value of 0x9495D388 and *SPDU_ID_3* has the value of 0x87F13E11.

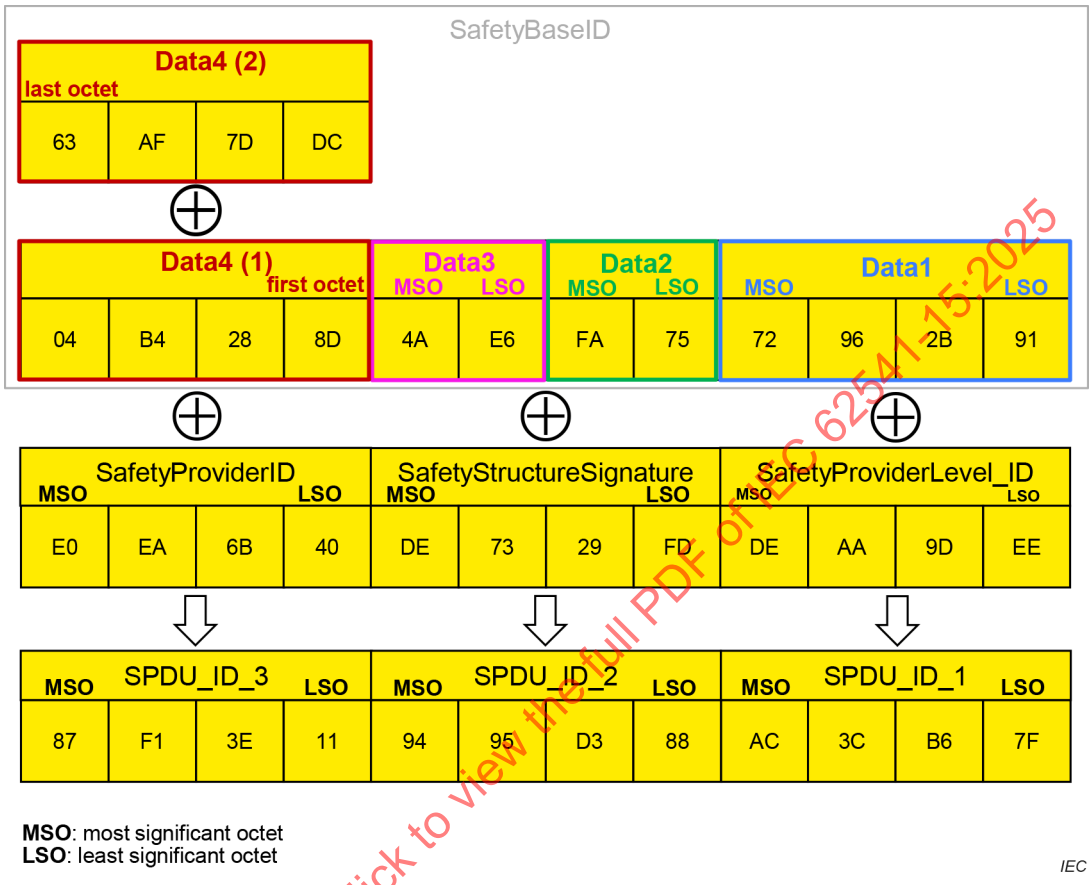


Figure 22 – Example for the calculation of SPDU_ID_1, SPDU_ID_2 and SPDU_ID_3

7.2.3.4 Coding of the SafetyProviderLevel_ID

The *SafetyProviderLevel* is the *SIL* the *SafetyProvider* implementation (hardware and software) is capable of.

Table 37 – Coding for the SafetyProviderLevel_ID

SafetyProviderLevel	Value of SafetyProviderLevel_ID
SIL1	0x11912881
SIL2	0x647C4654
SIL3	0xDEAA9DEE
SIL4	0xAB47F33B

[RQ7.17] Exactly one of the values provided in Table 37 shall be used as constant code value for *SafetyProviderLevel_ID*. The values were chosen in such a way that the hamming distance between them becomes maximal (hamming distance of 21).

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

[RQ7.18] Measures shall be taken to avoid that a *SafetyProvider* is erroneously using a code value belonging to a *SIL* that is higher than the *SIL* it is capable of. For instance, a *SafetyProvider* capable of *SIL*1-3 should not be able to accidently use the value 0xAB47F33B used for *SIL*4. One way to achieve this is to avoid that this constant appears in the source code of the *SafetyProvider* at all.

The *SafetyProviderLevel* is independent to the *SIL* capability of the provided *SafetyData*, see 3.1.2.12.

7.2.3.5 Signature over the SafetyData Structure (SafetyStructureSignature)

SafetyStructureSignature is used to check the number, *DataTypes*, and order of application data transmitted in *SafetyData*. If the *SafetyConsumer* is expecting anything different than what the *SafetyProvider* actually provides, *SafetyStructureSignature* will differ, allowing the *SafetyConsumer* to enable *fail-safe substitute values*.

In addition, the identifier of the *Structure DataType* (*SafetyStructureIdentifier*) is also taken into account when calculating *SafetyStructureSignature*. This ensures that the *SafetyProvider* and the *SafetyConsumer* are using the same identifier for the *Structure DataType*, effectively avoiding any confusion.

For instance, if a *SafetyProvider* defines a *Structure* with identifier "vec3D_m" comprising three *Floats* containing a three-dimensional vector in the metric system, this *Structure* could not be used by a *SafetyConsumer* expecting a *Structure* of *DataType* "vec3D_in" where the vector components are given in inches, or even at a *SafetyConsumer* expecting a *Structure* of *DataType* "orientation", containing three *Floats* to define an orientation using Euler angles.

[RQ7.19] *SafetyStructureSignature* shall be calculated as a 32 bit *CRC* signature (polynomial: *0x*F4ACFB13, see Clause B.1) over *SafetyStructureIdentifier* (encoding: UTF-8), *SafetyStructureSignatureVersion* and the sequence of the *DataTypeEncodingIDs*. After each *DataTypeEncodingID*, a 16-bit zero-value (0x0000) shall be inserted. All integers shall be encoded using little endian octet ordering. Data shall be processed in reverse order, see Clause B.1. The value "0" shall not be used as a signature. Instead, the value "1" shall be used in this case.

The terminating zero of *SafetyStructureIdentifier* shall not be considered when calculating the *CRC*.

[RQ7.20] *SafetyStructureIdentifier* may be visible in the IEC 62541 *Information Model* for diagnostic purposes but shall not be evaluated by the *SafetyConsumer* during runtime.

[RQ7.21] For all releases up to Release 2.0 of the specification, the value for *SafetyStructureSignatureVersion* shall be 0x0001.

Example:

SafetyStructureIdentifier,
e.g. "Motörhead" = 0x4d 0x6f 0x74 0xc3 0xb6 0x72 0x68 0x65 0x61 0x64

SafetyStructureSignatureVersion:= 0x0001

1. *DataType Int16*: (*DataTypeEncodingId* = 0x0004), // see 6.2.5

2. *DataType Boolean*: (*DataTypeEncodingId* = 0x0001),

3. *DataType Float*: (*DataTypeEncodingId* =0x000A)

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

SafetyStructureSignature =
= CRC32_Backward(0x4d, 0x6f, 0x74, 0xc3, 0xb6, 0x72, 0x68, 0x65, 0x61, 0x64,

0x01,0x00,

0x04,0x00, 0x00,0x00,

0x01,0x00, 0x00,0x00,

0x0A, 0x00, 0x00, 0x00) =

= CRC32_Forward(

0x00, 0x00, 0x00, 0x0A,

0x00, 0x00, 0x00, 0x01,

0x00, 0x00, 0x00, 0x04,

0x00,0x01,

0x64,0x61,0x65,0x68,0x72,0xb6,0xc3,0x74,0x6f,0x4d)

= 0xe2e86173

NOTE 1 The insertion of 0x0000 values after each *DataTypeEncodingID* allows for introducing arrays in later versions of this document.

NOTE 2 *SafetyStructureSignatureVersion* is the version of the procedure used to calculate the signature, as defined in requirement RQ7.19. If future releases of this document define an alternative procedure, they will indicate this by using a different version number.

IEC 62541-3:2020, 5.8.2 defines different categories of *DataTypes*. Regarding the *DataTypeEncodingID* which is to be used within the *SafetyStructureSignature*, the following holds:

- For *Built-in DataTypes*, the ID from Table 1 of IEC 62541-6:2020 is used as *DataTypeEncodingID*.
- For *Simple DataTypes*, the *DataTypeEncodingID* of the *Built-in DataType* from which they are derived is used.
- As of now, *Structured DataTypes* (including *OptionSets*) shall not be used within *SafetyData*. *Arrays* are not supported. Instead, multiple variables of the same type are used.
- *Enumeration DataTypes* are encoded on the wire as *Int32* and therefore shall use the *DataTypeEncodingID* of the *Int32 Built-in DataType*.

7.2.3.6 Calculation of a CRC signature

The *SafetyProvider* calculates the *CRC* signature (*ResponseSPDU.CRC*) and sends it to the *SafetyConsumer* as part of the *ResponseSPDU*. This enables the *SafetyConsumer* to check the correctness of the *ResponseSPDU* including the *SafetyData*, *flags*, *MNR*, *SafetyConsumerID* and *SPDU_ID* by recalculating the *CRC* signature (CRC_calc).

[RQ7.22] The generator polynomial *0x*F4ACFB13 shall be used for the 32-Bit *CRC* signature.

[RQ7.23] If *SafetyData* is longer than one octet (e.g. if it is of *DataType UInt16*, *Int16* or *Float*), it shall be decoded and encoded using little endian order in which the least significant octet appears first in the incremental memory address stream.

[RQ7.24] The calculation sequence shall begin with the highest memory address (n) of the STrailer counting back to the lowest memory address (0) and then include also the *SafetyData* beginning with the highest memory address.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Figure 23 shows the calculation sequence of a *ResponseSPDU CRC* on a little-endian machine, using an example *SafetyData* with the following fields:

Int32 var1
UInt32 var2
UInt16 var3
Int16 var4
Boolean var5

For the example given above, the Strailer (without *CRC*) and *SafetyData* have a combined length of 34 octets (16 octets STrailer without *CRC*, 12 octets of *SafetyData*). The calculation of *ResponseSPDU.CRC* (*SafetyProvider*) or CRC_calc (*SafetyConsumer*) is done in reverse order, from bottom to top. In the example shown in Figure 23, *CRC* calculation starts at octet index 33 (most significant octet of the *MNR*) and ends at octet index 0.

NOTE The reverse order ensures that the effectiveness of the *CRC* mechanism remains independent of any *CRCs* used within the underlying IEC 62541 channel, even if it would coincidentally use the same *CRC* polynomial.

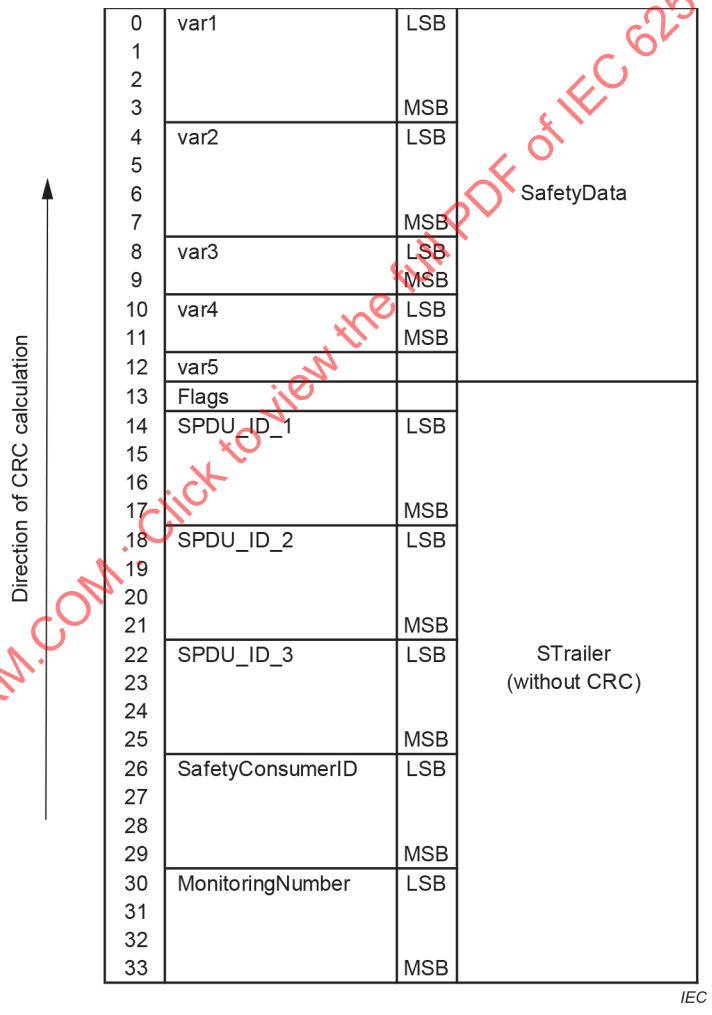


Figure 23 – Calculation of the CRC (on little-endian machines, CRC32_Backward)

An alternative way to calculate the *CRC* (particularly useful on big-endian machines) is shown in Figure 24. Here, the individual elements of the *ResponseSPDU* are already arranged in memory in reversed order, and *CRC* calculation is executed from octet 0 to octet 33.

See B.1 for examples for CRC calculation algorithms using tables.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

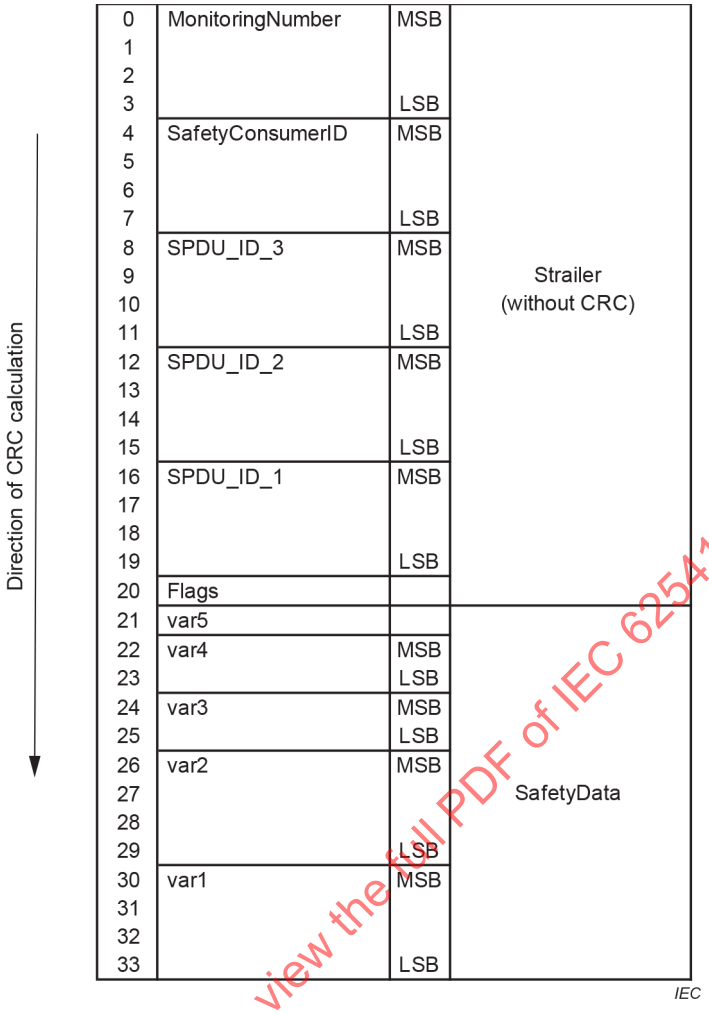


Figure 24 – Calculation of the CRC (on big-endian machines, CRC32_Forward)

[RQ7.25] On the *SafetyConsumer*, CRC_calc shall be calculated using data received in the *ResponseSPDU*, and not from expected values.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

8 Safety communication layer management

8.1 General

This clause gives details about the management of the *safety communication layer*.

8.2 Safety function response time part of communication

For cyclic communication, the part of the *safety function response time* attributable to a safety communication according to this document ($SFRT_{OPCSafety}$) is specified in Formula (1).

Calculation of safety function response time part of OPC UA safety

$$SFRT_{OPCSafety} <= 2 \times SafetyConsumerTimeout + ConsumerCycleTime \quad (1)$$

where

$SFRT_{OPCSafety}$: Part of the *safety function response time* attributable to the safety communication according to this document.

SafetyConsumerTimeout: Watchdog timer running in the *SafetyConsumer*. It is started immediately before a new *RequestSPDU* is sent (T14 or T28). If the timer runs out a timeout error is triggered (T18 or T29).

ConsumerCycleTime: The maximum time for the cyclic execution of the *SafetyConsumer*, see 7.2.2.2.

NOTE 1 Formula (1) only addresses the part of the *SFRT* attributable to OPC UA Safety. The overall *SFRT* also depends on the implementation of the devices the *SafetyProvider* and *SafetyConsumer* are running on. Details on how these fractions of the *SFRT* are calculated are vendor-specific.

If multiple safety connections according to this document are used within a safety function in series, their respective attributions to the *SFRT* shall be summed up.

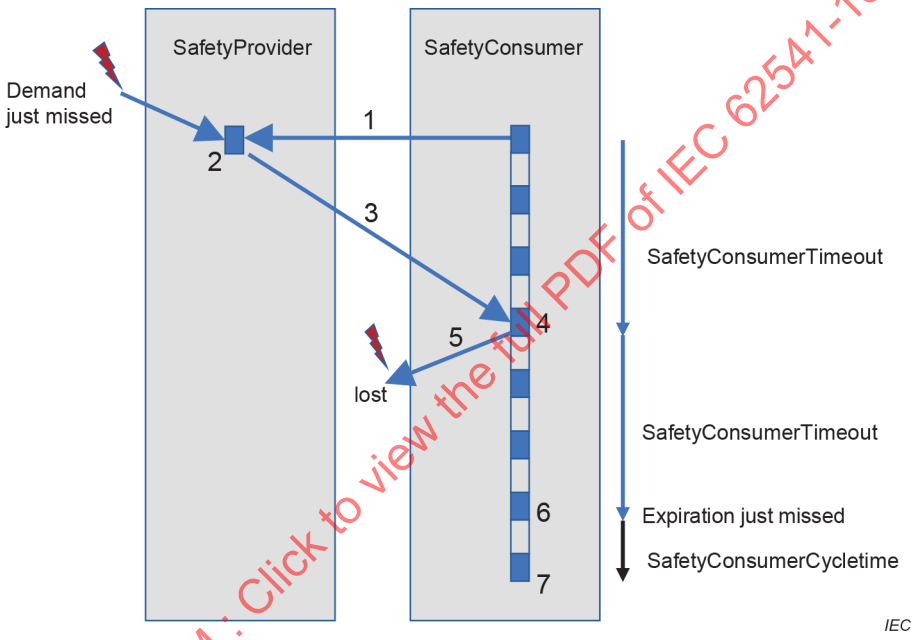


Figure 25 – Overview of delay times and watchdogs

Formula (1) is justified by Figure 25 and the following explanation:

1) The *SafetyConsumer* sends a *RequestSPDU*. At about the same time, a dangerous event occurs at the *SafetyProvider*, demanding the safety function to trigger.
2) However, in the worst case, the *RequestSPDU* is processed at the *SafetyProvider* just before the dangerous event becomes known.
3) Hence, the *ResponseSPDU* does not yet contain any information about the dangerous event.
4) In the worst case, the *ResponseSPDU* is processed in the *SafetyConsumer* just before the *SafetyConsumerTimeout* expires.
5) An error leads to a loss or unacceptable delay of either the *RequestSPDU* or the *ResponseSPDU*.
6) Hence, the *SafetyConsumerTimeout* expires.
7) In the worst case, the timer expires immediately after it was checked. Hence, it takes another cycle of the *SafetyConsumer* to detect the error.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

SafetyConsumerTimeout is a parameter of the *SafetyConsumer*. *ConsumerCycleTime* depends on the maximum sample time of the *SafetyConsumer* application. At commissioning, the integrator should be advised to design it shorter than a quarter of the target $SFRT_{OPCSafety}$. If the watchdog time *SafetyConsumerTimeout* is too small, spurious trips can occur. To avoid this, *SafetyConsumerTimeout* should be chosen as shown in Formula (2).

Selection of the watchdog parameter SafetyConsumerTimeout

$$\begin{aligned} \text{SafetyConsumerTimeout} >= {} & \text{T\_CD\_RequestSPDU} + \\ & \text{SafetyProviderDelay} + \\ & \text{T\_CD\_ResponseSPDU} + \\ & \text{SafetyConsumerDelay} \end{aligned} \tag{2}$$

where

T_CD_RequestSPDU: The worst-case communication delay for the *RequestSPDU*.

NOTE 2 T_CD_RequestSPDU can include the occurrence of dropped messages in the *standard transmission system*.

T_CD_ResponseSPDU: The worst-case communication delay for the *ResponseSPDU*.

NOTE 3 T_CD_ResponseSPDU can include the occurrence of dropped messages in the *standard transmission system*.

SafetyProviderDelay: The worst-case *SafetyProvider* delay.
Typically, one scan time period of the *SafetyProvider*.

SafetyConsumerDelay: The worst-case *SafetyConsumer* delay.
Typically, one scan time period of the *SafetyConsumer*.

NOTE 4 The reason why SafetyConsumerDelay is part of the summation is that it can take one cycle after the asynchronous reception of the *ResponseSPDU* to execute the checks.

[RQ8.1] To support the calculation of *SafetyConsumerTimeout* the *SafetyProvider* shall provide the *SafetyProviderDelay* as a *Variable* in the IEC 62541 *Information Model*, see Table 12.

Vendors may provide their individual adapted calculation method if necessary.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

9 System requirements (SafetyProvider and SafetyConsumer)

9.1 Constraints on the SPDU parameters

9.1.1 SafetyBaseID and SafetyProviderID

The pair of *SafetyProviderID* and *SafetyBaseID* is used by the *SafetyConsumer* to check the authenticity of the *ResponseSPDU*. *SafetyProviderID* and *SafetyBaseID* are usually assigned during engineering or during commissioning. It is in the responsibility of the end user or OEM to assign unique *SafetyProviderID* to individual *SafetyProviders* whenever this is reasonable possible. For instance, a machine builder should assign unique *SafetyProviderIDs* within a single machine containing multiple devices which run implementations of this document.

As the effort for the administration of unique *SafetyProviderIDs* will reach its limits when the system becomes large, this document uses the *SafetyBaseID* for cases where guaranteeing unique *SafetyProviderIDs* is not possible.

A *SafetyBaseID* is a universal unique identifier version4 (UUIDv4, also called *globally unique identifier* (*GUID*)), as described in ISO/IEC 9834-8:2014, Clause 15. It is a 128-bit number where at least 96 bits were chosen randomly. The probability that two randomly generated UUIDs are identical is extremely low ($2^{-96} < 10^{-28}$), and can therefore be neglected, even when considering applications with a *safety integrity level* of 4.

It is not necessary to generate an individual *SafetyBaseIDs* for all *SafetyProviders*. If two *SafetyProviders* can be discriminated by their *SafetyProviderIDs*, they may share the same *SafetyBaseID*. For instance, a machine builder could generate a unique *SafetyBaseID* for each instance of a machine, which is reused for all *SafetyProviders* within a machine.

When implementing or using a generator for the UUIDs, it shall be ensured that each possible value is generated with equal probability (discrete uniform distribution), and that any two values are independent from each other. When a pseudo random number generator (PNRG) is used, it is 'seeded' with a random source having enough collision entropy (e.g. seeds of at least 128 bits that are uniformly distributed, too; and all seeds being pairwise independent from each other).

Most commercial systems offer random number generators for applications within a cryptographic context. These applications pose even harder requirements on the quality of random numbers than the ones mentioned above. Hence, cryptographically strong random number generators are applicable to this document as well. See References [2] to [5], as well as IEC 62541-2, for detailed information.

Table 38 shows implementations of cryptographically strong random number-generators that can be used to calculate the random part of the UUIDv4:

Table 38 – Examples for cryptographically strong random number generators

Environment	Function
Microsoft® Windows® Operating Systems	BcryptGenRandom found in Bcrypt.dll
Unix®-like OS (e.g. Linux® / FreeBSD® / Solaris®)	Read from the file: /dev/urandom/
.NET®	RandomNumberGenerator from System.Security.Cryptography
JavaScript®	Crypto.getRandomValues()
Java®	java.security.SecureRandom
Python®	os.urandom(size)

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

While being evaluated from a security point of view, probably none of these implementations has been validated with safety in mind. Therefore, there is a remaining risk that these implementations are subject to systematic implementation errors which could decrease the effectiveness of these random numbers. To overcome this problem, the output of the random number generator is not used directly, but a SHA256-hash is calculated over (1) the generator's output, (2) a timestamp (wall-clock-time or persistent logical clock) and (3) a unique domain name. Any bits of the SHA256-hash can then be used to construct the random parts of the UUIDv4.

[RQ9.1] The parameters *SafetyBaseID* and *SafetyProviderID* shall be stored in a non-volatile, i.e. persistent, way.

9.1.2 SafetyConsumerID

The *SafetyConsumerID* allows for discrimination between *RequestSPDUs* and *ResponseSPDUs* belonging to different *SafetyConsumers*. It is mainly used for diagnostic purposes, such as detecting unintentional concurrent access of a single *SafetyProvider* by multiple *SafetyConsumers*. *Safety-related* communication errors which are detected by checking the *SafetyConsumerID* would also be detected by other mechanisms, including the *MNR*, the *SafetyProviderID*, and the *SafetyConsumerTimeout*.

From a safety point of view, there are no qualitative requirements regarding the generation or administration of the *SafetyConsumerID*. It may be assigned during engineering, commissioning, at startup, and may even change during runtime. It is not required to check for uniqueness of *SafetyConsumerIDs*.

However, assigning identical *SafetyConsumerIDs* to multiple consumers is not recommended because fault localization can become more difficult.

9.2 Initialization of the MNR in the SafetyConsumer

The *MNR* is used to discriminate messages stemming from the same *SafetyProvider* and is therefore used to detect timeliness errors such as outdated messages, messages received out-of-order, or streams of messages erroneously repeated by a network storing element (e.g. a router).

To be effective, the set of used *MNR* values shall not be restricted to a small set. This could happen for connections which are restarted frequently, and which start counting from the same *MNR* value each time.

There are at least two ways to address this potential problem:

Option 1: [RQ9.2a] Whenever the connection is terminated, the current value of the *MNR* shall be safely stored within non-volatile memory of the *SafetyConsumer*. After restart, the previously stored *MNR* is used for initialization of the *MNR* (i.e. in state S12 of the *SafetyConsumer* state machine).

Option 2: [RQ9.2b] Whenever the *SafetyConsumer* is restarted (i.e. in state S12 of the *SafetyConsumer* state machine), the *MNR* is initialized with a 32-bit random number.

Either requirement RQ9.2a or requirement RQ9.2b, or an equivalent solution shall be fulfilled.

9.3 Constraints on the calculation of system characteristics

9.3.1 Probabilistic considerations (informative)

Following IEC 61784-3, this document detects all communication errors which can possibly occur in the underlying *standard transmission system*, including the IEC 62541 stack. If an error is detected, the erroneous data is discarded. Moreover, this document is designed in such a way that a safety function becomes practically unusable if the failure rate in the underlying, *standard transmission system* is higher than one error per safety error interval limit (6 min, 60 min, or 600 min), depending on the desired *SIL* of the safety function (see Table 26 and Table 39).

Thus, for operational safety functions a failure rate of 0,1 h^{-1}, 1 h^{-1}, or 10 h^{-1} can be assumed for communication errors occurring in the IEC 62541 stack. In order to obtain the communication's contribution to the PFH value of the safety function, this value has to be multiplied by the so-called conditional residual error probability $P_{re,cond}$. For the *CRC* mechanism used in this document, it holds:

$$P_{re,cond} \leq 4{,}0 \times 10^{-10}$$

This leads to the PFH and PFD values shown in Table 39.

The value 4,0 × 10^{-10} was justified by extensive numerical evaluation of the 32-bit *CRC* generator polynomial in use (*0x*F4ACFB13). The results of this evaluation, executed for all relevant data lengths and all relevant values for the bit error probability p up to p = 0,5, is shown in Figure 26. As can be seen, $P_{re,cond}$ never exceeds the value 4,0 × 10^{-10}.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

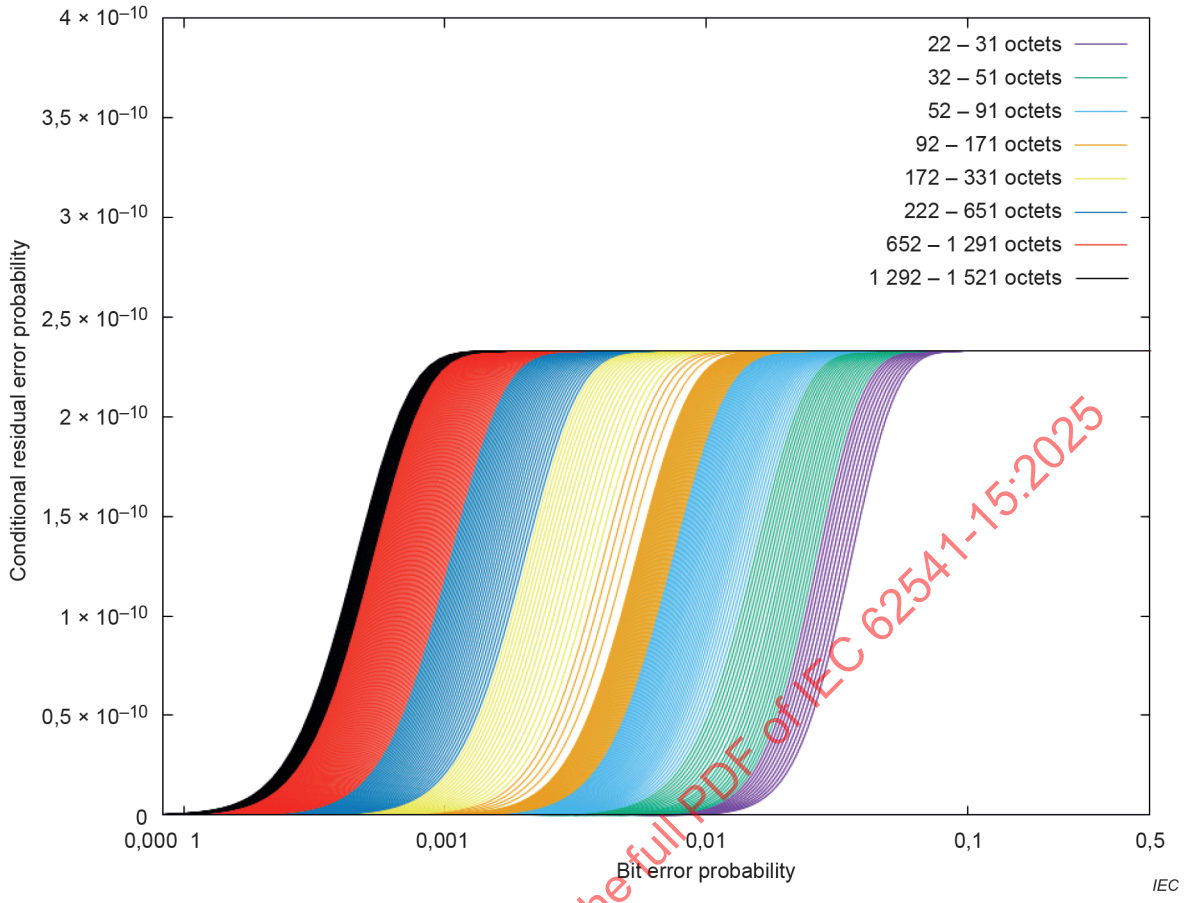


Figure 26 – Conditional residual error probability of the CRC check

An explanation that it is indeed necessary to calculate $P_{\text{re,cond}}$ for all data lengths and all relevant values of p can be found in Figure 27. For the data lengths shown in this figure, $P_{\text{re,cond}}$ exceeds the desired value by several orders of magnitudes. The maximum value of $P_{\text{re,cond}}$ is not obtained when p becomes maximal.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

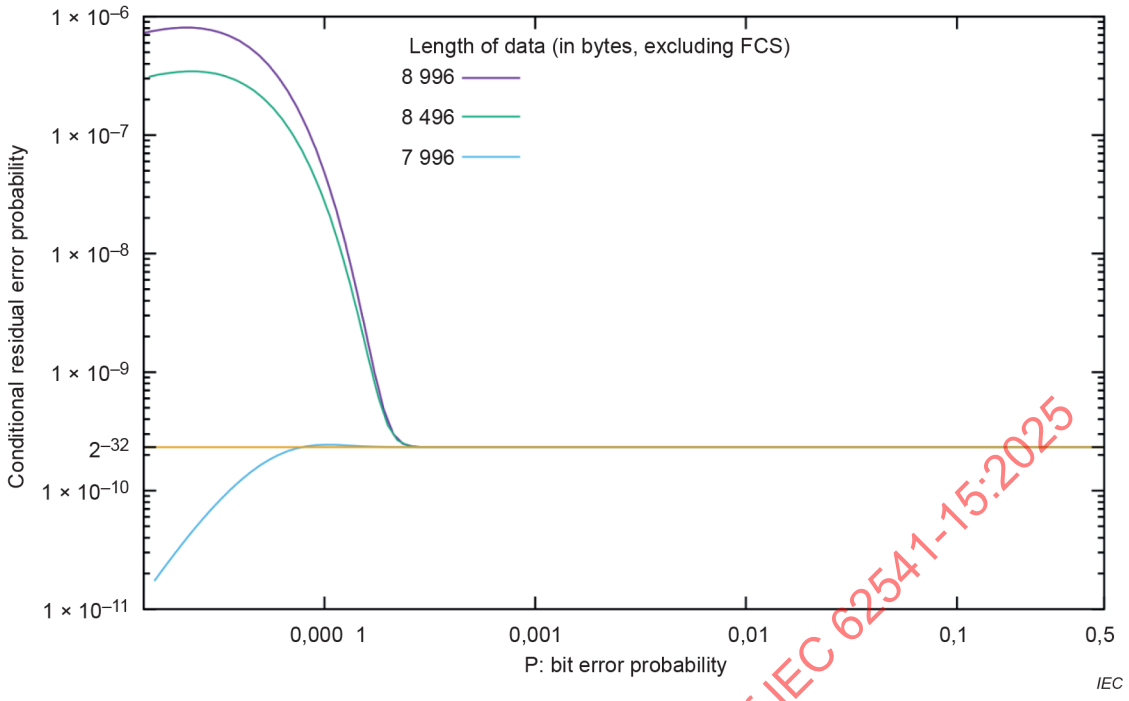


Figure 27 – Counter example: data lengths not supported by OPC Safety

9.3.2 Safety related assumptions (informative)

The boundary conditions and assumptions for safety assessments and calculations of *residual error rates* are listed here.

Generally:

- Number of retries in the underlying *standard transmission system*:
  No restrictions
- *CRC* polynomials used inside the underlying *standard transmission system*(e.g. Ethernet, TCP, …):
  No restrictions
- Message storing elements:
  No restrictions, any number of message storing elements is permitted
- Size of *SafetyData* within one *ResponseSPDU*:
  ≤1 500 octets

Even for safety functions that do not require manual operator acknowledgment for restart, manual operator acknowledgment is mandatory whenever the *SafetyConsumer* has detected certain types of errors and indicates this using *OperatorAckRequested*. Hence, operator acknowledgment is expected to be implemented by the safety application whenever OPC UA Safety is used. For details, see 6.3.4.3 and Clause B.2.

9.4 PFH and PFD values of a logical safety communication link

The PFH value of a logical safety communication link according to this document depends on the parameter of *SafetyErrorIntervalLimit* (see Table 26) of the link's *SafetyConsumer*. Whenever the *SafetyConsumer* detects a mismatch of the *SafetyConsumerID*, *SPDU_ID*, *MNR* or *CRC*, it will only continue operating if the last occurrence of such an error happened more than *SafetyErrorIntervalLimit* time units ago. Otherwise, it will make a transition to *fail-safe* values, which can only be left by manual operator acknowledgment, see 6.3.4.3.

This directly limits the rate of detected errors, and indirectly limits the rate of undetected (residual) errors.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

See Table 39 for numeric PFH and PFD values.

Table 39 – The total residual error rate for the safety communication channel

SafetyErrorIntervalLimit	Allowed for SIL range	Total residual error rate for one logical connection of the safety function (PFH)	Total residual error probability for one logical connection of the safety function, for a mission time of 20 years (PFDavg)
6 min	Up to *SIL*2	$< 4{,}0 \times 10^{-9}$ / h	$< 1{,}0 \times 10^{-6}$
60 min	Up to *SIL*3	$< 4{,}0 \times 10^{-10}$ / h	$< 2{,}5 \times 10^{-7}$
600 min	Up to *SIL*4	$< 4{,}0 \times 10^{-11}$ / h	$< 8{,}0 \times 10^{-8}$

The parameter *SafetyErrorIntervalLimit* affects either the PFH or the PFD, or both of only the *safety communication channel*. There is no effect on the PFH and PFD values of the components the *SafetyProviders* and *SafetyConsumers* are running on. The requirements for the implementation of these components are specified in the IEC 61508 series.

9.5 Safety manual

[RQ9.3] According to IEC 61508-2, the suppliers of equipment implementing an implementation of this document shall provide a safety manual. The instructions, information and parameters of Table 40 shall be included in that safety manual unless they are not relevant for a specific device.

Table 40 – Information to be included in the safety manual

	Item	Instruction or parameter	Remark
1	Safety handling	Instructions on how to configure, parameterize, commission and test the device safely in accordance with the IEC 61508 series and IEC 61784-3.	
2	PFH, respectively PFDavg	The PFH, respectively PFDavg, per logical connection of the safety function.	See 9.3.2 and 9.4
3	$SFRT_{OPCSafety}$	Information on how this value can be calculated by the end user or OEM.	See 8.1 The implementation and error reaction of *ConsumerCycleTime* is in the responsibility of the either the vendor or the integrator, or both.
4	*SafetyBaseID* / *SafetyProviderID*	Information on how the *SafetyBaseID* and *SafetyProviderID* are generated and assigned.	See 9.1.1
5	Commissioning	Either the end user or the OEM, or both, are responsible for verification and validation of correct cabling and assignment of network addresses. The safety manual shall address how this can be accomplished.	

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

	Item	Instruction or parameter	Remark
6	Operator acknowledgment	If the *SafetyConsumers* makes a transition to *fail-safe substitute values* requiring operator acknowledgment "frequently", this is an indication that a check of the installation (for example electromagnetic interference), network traffic load, or transmission quality is required. It shall be mentioned in the manual that it is potentially unsafe to simply omit these checks. "Frequently" in this context is defined as – more than once per day in *SIL*2 and *SIL*3 applications – more than once per week in *SIL*4 applications	
7	High demand and low demand applications	The *SafetyConsumer* shall be executed cyclically within a shorter time frame than the *SafetyConsumerTimeout*.	
8	Maintenance	Specific requirements for device repair and device replacement.	
9	Relevant safety standards	A safety device according to this document shall fulfill the requirements of the relevant safety standards, such as the IEC 61508 series (according to the *SIL* as described) when used in live operation.	For usage in live operation.

9.6 Indicators and displays

[RQ9.4] The device a *SafetyConsumer* is running on shall be able to indicate if *SAPI.OperatorAckRequested* is enabled. This can be done for example by an indicator LED or by using an HMI.

[RQ9.5] If an LED is used for indication, it shall blink in green colour with frequency of 0,5 Hz whenever the output *SAPI.OperatorAckRequested* is true of at least one of the *SafetyConsumers* running on the device.

This LED may also be used for other purposes. For instance, normal operation may be indicated by a non-flashing LED, or erroneous behaviour may be indicated by an LED blinking with a frequency higher than 0,5 Hz. Thus, this document does not contain any requirements for the behaviour of the LED if *SAPI.OperatorAckRequested* is false.

The message shown on an HMI is application-specific. For instance, the text "machine has stopped for safety reasons. For restart, please check for obstacles and press the green button." can be shown.

NOTE How to realize operator acknowledgment (physical button, element in HMI etc.) is vendor-specific.

10 Assessment

10.1 Safety policy

Users of this document shall take into account the following constraints To avoid misunderstanding or wrong expectations regarding safety-related developments and applications.

NOTE 1 This includes for example use for training, seminars, workshops and consultancy.

The communication technologies specified in this document shall only be implemented in devices designed in accordance with the requirements of the relevant safety standards.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

The use of communication technologies specified in this document in a device does not ensure that all necessary technical, organizational and legal requirements related to safety-related applications of the device have been fulfilled in accordance with the requirements of the relevant safety standards.

For a device based on this document to be suitable for use in safety-related applications, appropriate functional safety management life-cycle processes according to the relevant safety standards shall be observed. This shall be assessed in accordance with the independence and competence requirements of the relevant safety standards. Safety-related applications of the device can be subject to local regulations and legal requirements.

NOTE 2 Examples for relevant safety standards include the IEC 61508 series, IEC 61511, IEC 60204-1, IEC 62061, ISO 13849-1 and ISO 13849-2.

The manufacturer of a device using communication technologies specified in this document is responsible for the correct implementation of the standard, the correctness and completeness of the device documentation and information.

Additional important information including corrigenda and errata published by IEC shall be considered for implementation and assessment.

It is strongly recommended that implementers of this document comply with the appropriate conformance tests and validations provided by the related technology-specific organization. Refer to Annex C for additional information about assessment and conformance validation.

NOTE 3 These requirements and recommendations are included because incorrect implementations could lead to serious injury or loss of life.

10.2 Obligations

Since safety technology in automation is relevant to occupational safety and the concomitant insurance risks in a country, local regulations and legal requirements can apply. The national authorities (notified bodies) decide on the recognition of assessment reports.

NOTE Examples of such authorities are the IFA (Institut für Arbeitsschutz der Deutschen Gesetzlichen Unfallversicherung/Institute for Occupational Safety and Health of the German Social Accident Insurance) in Germany, HSE (Health and Safety Executive) in UK, FM (Factory Mutual/Property Insurance and Risk Management Organization), UL (Underwriters Laboratories Inc./Product Safety Testing and Certification Organization), or the INRS (Institut National de Recherche et de Sécurité) in France.

10.3 Index of requirements (informative)

Table 41 gives an informative overview of all the requirements (safety and *non-safety*) which are described in this document. A summary requirement description and the corresponding clause or subclause where the requirement is defined are given. To fully understand a requirement and its context, it is necessary to consult its original definition. Table 41 serves as a tool for quick navigation and as a checklist for an overview over all requirements.

For the conventions used for numbering requirements, see 3.3.2.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Table 41 – Index of requirements (informative)

Requirement number	Requirement summary	Clause or subclause
RQ4.1	Implement in devices designed according to the IEC 61508 series with appropriate SIL	4.2 Implementation aspects
RQ5.1	Implement in safety devices only	5.2 Safety functional requirements
RQ5.2	Implement *safety measures* (*MNR*, timeout with receipt, IDs, data integrity check)	5.3 Safety measures
RQ5.3	Process and monitor *safety measures* in the *SCL*	5.3 Safety measures
RQ5.4	Start *CRC* calculation with value "1"	5.5 Requirements for CRC calculation
RQ5.5	Use *CRC* result "1" instead of "0"	5.5 Requirements for CRC calculation
RQ5.6	Ignore all-zero *SPDUs*	5.5 Requirements for CRC calculation
RQ6.1	Singleton *SafetyACSet Folder*	6.2.2.1 SafetyACSet Object
RQ6.2	*Objects* for *SafetyProviders* and *SafetyConsumers*	6.2.2.1 SafetyACSet Object
RQ6.3a	Usage of *Call Service* for *Client/Server*	6.2.2.1 SafetyACSet Object
RQ6.3b	Usage of *SafetyPDUs* for *PubSub*	6.2.2.1 SafetyACSet Object
RQ6.4	Provide *SPDUs* for diagnostics in *Method ReadSafetyDiagnostics*	6.2.2.1 SafetyACSet Object
RQ6.5	Restrictions on *DataTypes*	6.2.2.2 Safety ObjectType definitions
RQ6.6	Non-abstract *DataTypes* for out data	6.2.2.2 Safety ObjectType definitions
RQ6.7	Definition of concrete *DataTypes* for *ResponseSPDU*	6.2.3.4 ResponseSPDUDataType
RQ6.8	Usage of *NonSafetyDataPlaceHolder*	6.2.3.4 ResponseSPDUDataType
RQ6.9	Restriction to scalar types	6.2.5 DataTypes and length of SafetyData
RQ6.10	List supported *DataTypes* in user manual	6.2.5 DataTypes and length of SafetyData
RQ6.11	Values for *Boolean DataType*	6.2.5 DataTypes and length of SafetyData
RQ6.12	Implementation of *SafetyProvider SAPI*	6.3.3.2 SAPI of SafetyProvider
RQ6.13a	Implementation of *SafetyProvider SPI*	6.3.3.3 For additional information regarding operator acknowledgment use cases, see Clause B.3. SPI of SafetyProvider
RQ6.13b	Parameters of *SafetyProvider SPI*	6.3.3.3 For additional information regarding operator acknowledgment use cases, see Clause B.3. SPI of SafetyProvider
RQ6.14	Implementation of *SafetyConsumer SAPI*	6.3.4.2 SAPI of SafetyConsumer
RQ6.15a	Implementation of *SafetyConsumer SPI*	6.3.4.4 SPI of the SafetyConsumer
RQ6.15b	Parameters of *SafetyConsumer SPI*	6.3.4.4 SPI of the SafetyConsumer
RQ6.16	Values for *qualifiers*	6.3.6 Principle for "application variables with qualifier"
RQ6.17	*SafetyConsumer* diagnostic message texts	6.4.2 Diagnostics messages of the SafetyConsumer
RQ7.1	*RequestSPDU* Flags	7.2.1.4 RequestSPDU: Flags
RQ7.2	Contents and structure of *SafetyData* in *ResponseSPDU*	7.2.1.5 ResponseSPDU: SafetyData
RQ7.3	Usage of *ResponseSPDU.Flags*	7.2.1.6 ResponseSPDU: Flags
RQ7.4	Zero out reserved *flags*	7.2.1.6 ResponseSPDU: Flags
RQ7.5	Copy *SafetyConsumerID* into *ResponseSPDU*	7.2.1.8 ResponseSPDU: SafetyConsumerID

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Requirement number	Requirement summary	Clause or subclause
RQ7.6	Copy *MonitoringNumber* into *ResponseSPDU*	7.2.1.9 ResponseSPDU: MonitoringNumber
RQ7.7	Usage of *CRC* signature	7.2.1.10 ResponseSPDU: CRC
RQ7.8	Usage of *NonSafetyData*	7.2.1.11 ResponseSPDU: NonSafetyData
RQ7.9	Indication of *NonSafetyData*	7.2.1.11 ResponseSPDU: NonSafetyData
RQ7.10	Answer repeated *RequestSPDUs* in *Client/Server* communication	7.2.2.2 SafetyProvider and SafetyConsumer Sequence diagram
RQ7.11	Document behaviour chosen in RQ7.10 in safety manual	7.2.2.2 SafetyProvider and SafetyConsumer Sequence diagram
RQ7.12	Monitor *ConsumerCycleTime* in safety-related way	7.2.2.2 SafetyProvider and SafetyConsumer Sequence diagram
RQ7.13	Implement *SafetyProvider* behaviour	7.2.2.4 SafetyProvider state diagram
RQ7.14	Implement *SafetyConsumer* behaviour	7.2.2.5 SafetyConsumer state diagram
RQ7.15	Rules for building the *ResponseSPDU*	7.2.3.1 Build ResponseSPDU
RQ7.16	Rules for calculating *SPDU_ID* fields	7.2.3.2 Calculation of the SPDU_ID_1, SPDU_ID_2, SPDU_ID_3
RQ7.17	Values to indicate *SafetyProviderLevel_ID*	7.2.3.4 Coding of the SafetyProviderLevel_ID
RQ7.18	Avoid accidental use of higher SIL indicator	7.2.3.4 Coding of the SafetyProviderLevel_ID
RQ7.19	Calculation of *SafetyStructureSignature*	7.2.3.5 Signature over the SafetyData Structure (SafetyStructureSignature)
RQ7.20	No evaluation of *SafetyStructureSignature*	7.2.3.5 Signature over the SafetyData Structure (SafetyStructureSignature)
RQ7.21	Value of *SafetyStructureSignatureVersion*	7.2.3.5 Signature over the SafetyData Structure (SafetyStructureSignature)
RQ7.22	Generator polynomial for *CRC* signature	7.2.3.6 Calculation of a CRC
RQ7.23	Endianess encoding of *SafetyData*	7.2.3.6 Calculation of a CRC
RQ7.24	*CRC* calculation sequence	7.2.3.6 Calculation of a CRC
RQ7.25	Calculate *CRC* in *SafetyConsumer* from *ResponseSPDU* values	7.2.3.6 Calculation of a CRC
RQ7.26	Immediate effect of *SafetyConsumerTimeout*	7.2.2.2 SafetyProvider and SafetyConsumer Sequence diagram
RQ8.1	Provision of SafetyProviderDelay	8.2 Safety function response time part of communication
RQ9.1	Storage of *SafetyBaseID* and *SafetyProviderID*	9.1.1 SafetyBaseID and SafetyProviderID
RQ9.2a	(Option 1) Use stored *MNR* after restart	9.2 Initialization of the MNR in the SafetyConsumer
RQ9.2b	(Option 2) Use random *MNR* after restart	9.2 Initialization of the MNR in the SafetyConsumer
RQ9.3	Provision of and information in safety manual	9.5 Safety manual
RQ9.4	Indication of *SAPI.OperatorAckRequested*	9.6 Indicators and displays
RQ9.5	Properties of LED indication of *SAPI.OperatorAckRequested*	9.6 Indicators and displays
RQ12.1	Namespaces	12.2 Handling of IEC 62541 namespaces

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

11 Profiles and conformance units

Figure 28 shows an informative overview of the *Facets* and *ConformanceUnits* pertaining to this document. For normative reference consult the *Profile Reporting* at https://profiles.opcfoundation.org/.

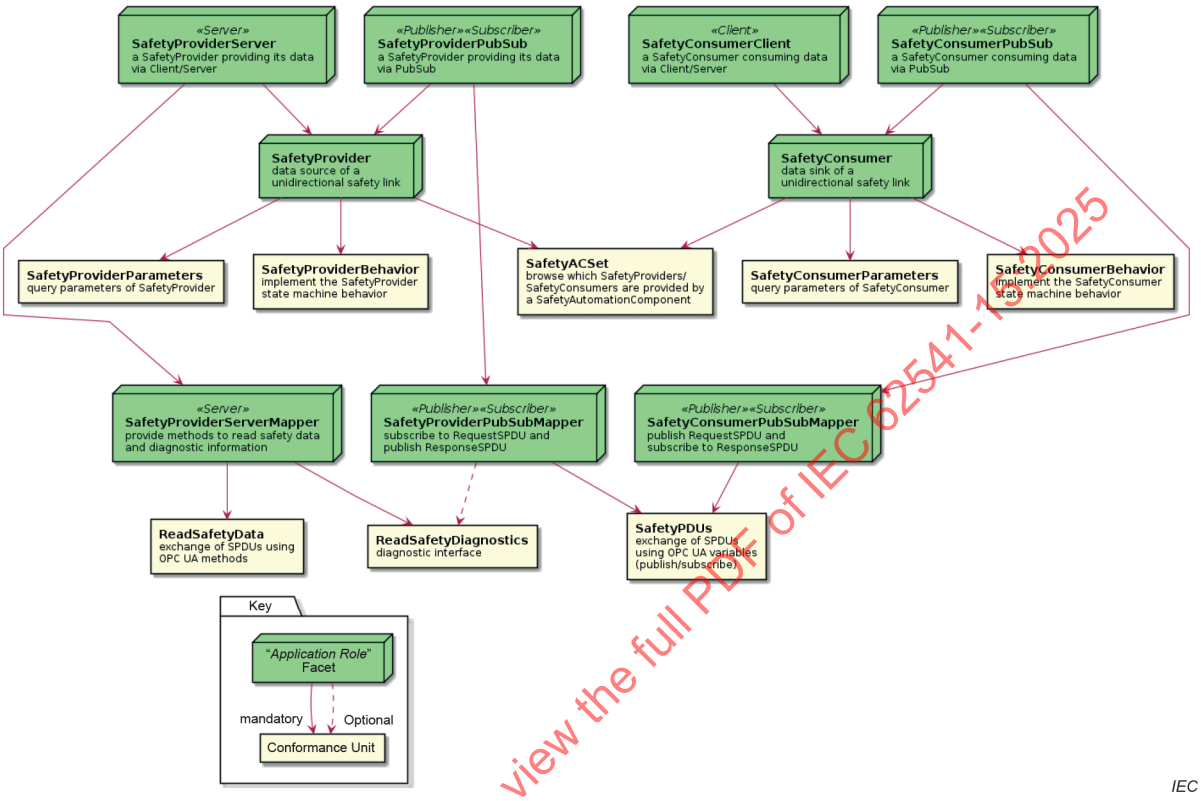


Figure 28 – Facets and ConformanceUnits

12 Namespaces

12.1 Namespace metadata

Table 42 defines the *Namespace* metadata for this document. The *Object* is used to provide version information for the *Namespace* and an indication about static *Nodes*. Static *Nodes* are identical for all *Attributes* in all *Servers*, including the *Value Attribute*. See IEC 62541-5 for more details.

The information is provided as *Object* of type *NamespaceMetadataType*. This *Object* is a component of the *Namespaces Object* that is part of the *Server Object*. The *NamespaceMetadataType ObjectType* and its *Properties* are defined in IEC 62541-5.

The version information is also provided as part of the *ModelTableEntry* in the *UANodeSet* XML file. The *UANodeSet* XML schema is defined in IEC 62541-6.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Table 42 – NamespaceMetadata Object for this document

Attribute	Value		
BrowseName	http://opcfoundation.org/UA/Safety		
Property	**DataType**	**Value**	
NamespaceUri	String	http://opcfoundation.org/UA/Safety	
NamespaceVersion	String	1.05.04	
NamespacePublicationDate	DateTime	2024-06-12	
IsNamespaceSubset	Boolean	False	
StaticNodeIdTypes	IdType []	0	
StaticNumericNodeIdRange	NumericRange []		
StaticStringNodeIdPattern	String		

12.2 Handling of IEC 62541 namespaces

Namespaces are used by IEC 62541 to create unique identifiers across different naming authorities. The *Attributes NodeId* and *BrowseName* are identifiers. A *Node* in the UA *AddressSpace* is unambiguously identified using a *NodeId*. Unlike *NodeIds*, the *BrowseName* cannot be used to unambiguously identify a *Node*. Different *Nodes* can have the same *BrowseName*. They are used to build a browse path between two *Nodes* or to define a standard *Property*.

Servers can often choose to use the same *Namespace* for the *NodeId* and the *BrowseName*. However, if they want to provide a standard *Property*, its *BrowseName* shall have the *Namespace* of the standards body although the *Namespace* of the *NodeId* reflects something else, for example the *EngineeringUnits Property*. All *NodeIds* of *Nodes* not defined in this document shall not use the standard namespaces.

[RQ12.1] Table 43 provides a list of mandatory and optional namespaces used in an IEC 62541 *Server* according to this document.

Refer to Annex A for details on *Namespaces*.

Table 43 – Namespaces used in a safety Server

NamespaceURI	Description	Use
http://opcfoundation.org/UA/	*Namespace* for *NodeIds* and *BrowseNames* defined in the IEC 62541 specification. This *Namespace* shall have *Namespace* index 0.	Mandatory
Local server URI	*Namespace* for Nodes defined in the local *Server*. This can include types and instances used in an AutoID Device represented by the *Server*. This *Namespace* shall have *Namespace* index 1.	Mandatory
http://opcfoundation.org/UA/Safety	*Namespace* for *NodeIds* and *BrowseNames* defined in this document. The *Namespace* index is *Server*-specific.	Mandatory
Vendor-specific types	A *Server* can provide vendor-specific types like types derived from *ObjectTypes* defined in this document in a vendor-specific *Namespace*.	Optional
Vendor-specific instances	A *Server* provides vendor-specific instances of the standard types or vendor-specific instances of vendor-specific types in a vendor-specific *Namespace*. It is recommended to separate vendor-specific types and vendor-specific instances into two or more namespaces.	Mandatory

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Annex A
(normative)

Safety namespace and mappings

This Annex A defines the numeric identifiers for the numeric *NodeIds* defined in the *Namespace* of this document. The identifiers are specified in a CSV file with the following syntax:

`<SymbolName>, <Identifier>, <NodeClass>`

Where the *SymbolName* is either the *BrowseName* of a *Type Node* or the *BrowsePath* for an *Instance Node* that appears in the specification and the *Identifier* is the numeric value for the *NodeId*.

The *NamespaceUri* for all *NodeIds* defined here is http://opcfoundation.org/UA/Safety

The CSV released with this version of the specification can be found here:

http://www.opcfoundation.org/UA/schemas/1.05/Opc.Ua.Safety.NodeIds.csv

The latest CSV that is compatible with this version of the specification can be found here:

http://www.opcfoundation.org/UA/schemas/Opc.Ua.Safety.NodeIds.csv

A computer processible version of the complete *Information Model* defined in this document is also provided. It follows the XML *Information Model* schema syntax defined in IEC 62541-6.

The *Information Model* schema released with this version of the specification can be found here:

http://www.opcfoundation.org/UA/schemas/1.05/Opc.Ua.Safety.NodeSet2.xml

The latest *Information Model* schema that is compatible with this version of the specification can be found here:

http://www.opcfoundation.org/UA/schemas/Opc.Ua.Safety.NodeSet2.xml

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Annex B
(informative)

Additional information

B.1 CRC calculation using tables, for the polynomial 0xF4ACFB13

The calculation of a 32-bit *CRC* signature over an array of N octets with the help of lookup tables, using "C" as programming language, is shown below:

```
// VARIANT A: presumably easier to implement on little endian machines
uint32_t crctab32[256];                                  // lookup table
uint32_t CRC32_Backward(char *array, int16_t N){         // input is array of N octets
                                                         // containing the data,
                                                         see Figure 23

    uint32_t result = 1;                    // seed value for the calculated CRC
    int16_t i;                              // index
    for(i=N-1;i>=0;i--)                     // process array in reversed order
        result = crctab32 [((result >> 24) ^ array[i]) & 0xff] ^ (result << 8);
    if (result==0)
            return 1;
    else
            return result;
}
```

where the lookup-table crctab32 has to be initialized as shown in Table B.1.

```
// VARIANT B: presumably easier to implement on big endian machines
uint32_t crctab32[256];                                  // lookup table
uint32_t CRC32_Forward(char *array, int16_t N){          // input is array of N octets
                                                         // containing the data in reversed
                                                         // order, see e. g. Figure 24

    uint32_t result = 1;                    // seed value for the calculated CRC
    int16_t i;                              // index
    for(i=0;i<N;i++)                        // process array
        result = crctab32 [((result >> 24) ^ array[i]) & 0xff] ^ (result << 8);
    if (result==0)
            return 1;
    else
            return result;
}
```

where the lookup-table crctab32 has to be initialized as shown in Table B.1.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Table B.1 – The CRC32 lookup table for 32-bit CRC signature calculations

CRC32 lookup table (0 to 255)							
00000000	F4ACFB13	1DF50D35	E959F626	3BEA1A6A	CF46E179	261F175F	D2B3EC4C
77D434D4	8378CFC7	6A2139E1	9E8DC2F2	4C3E2EBE	B892D5AD	51CB238B	A567D898
EFA869A8	1B0492BB	F25D649D	06F19F8E	D44273C2	20EE88D1	C9B77EF7	3D1B85E4
987C5D7C	6CD0A66F	85895049	7125AB5A	A3964716	573ABC05	BE634A23	4ACFB130
2BFC2843	DF50D350	36092576	C2A5DE65	10163229	E4BAC93A	0DE33F1C	F94FC40F
5C281C97	A884E784	41DD11A2	B571EAB1	67C206FD	936EFDEE	7A370BC8	8E9BF0DB
C45441EB	30F8BAF8	D9A14CDE	2D0DB7CD	FFBE5B81	0B12A092	E24B56B4	16E7ADA7
B380753F	472C8E2C	AE75780A	5AD98319	886A6F55	7CC69446	959F6260	61339973
57F85086	A354AB95	4A0D5DB3	BEA1A6A0	6C124AEC	98BEB1FF	71E747D9	854BBCCA
202C6452	D4809F41	3DD96967	C9759274	1BC67E38	EF6A852B	06337 0D	F29F881E
B850392E	4CFCC23D	A5A5341B	5109CF08	83BA2344	7716D857	9E4F2E71	6AE3D562
CF840DFA	3B28F6E9	D27100CF	26DDFBDC	F46E1790	00C2EC83	599B1AA5	1D37E1B6
7C0478C5	88A883D6	61F175F0	955D8EE3	47EE62AF	B34299BC	5A1B6F9A	AEB79489
0BD04C11	FF7CB702	16254124	E289BA37	303A567B	C496AD68	2DCF5B4E	D963A05D
93AC116D	6700EA7E	8E591C58	7AF5E74B	A8460B07	5CEAF014	B5B30632	411FFD21
E47825B9	10D4DEAA	F98D288C	0D21D39F	DF923FD3	2B3EC4C0	C26732E6	36CBC9F5
AFF0A10C	5B5C5A1F	B205AC39	46A9572A	941ABB66	60B64075	89EFB653	7D434D40
D82495D8	2C886ECB	C5D198ED	317D63FE	E3CE8FB2	176274A1	FE3B8287	0A977994
4058C8A4	B4F433B7	5DADC591	A9013E82	7BB2D2CE	8F1E29DD	6647DFFB	92EB24E8
378CFC70	C3200763	2A79F145	DED50A56	0C66E61A	F8CA1D09	1193EB2F	E53F103C
840C894F	70A0725C	99F9847A	6D557569	BFE69325	4B4A6836	A2139E10	56BF6503
F3D8BD9B	07744688	EE2DB0AE	1A814BBD	C832A7F1	3C9E5CE2	D5C7AAC4	216B51D7
6BA4E0E7	9F081BF4	7651EDD2	82FD16C1	504EFA8D	A4E2019E	4DBBF7B8	B9170CAB
1C70D433	E8DC2F20	0185D908	F5292215	279ACE59	D336354A	3A6FC36C	CEC3387F
F808F18A	0CA40A99	E5FDFCBF	115107AC	C3E2EBE0	374E10F3	DE17E6D5	2ABB1DC6
8FDCC55E	7B703E4D	9229C86B	66853378	B436DF34	409A2427	A9C3D201	5D6F2912
17A09822	E30C6331	0A559517	FEF96E04	2C4A8248	D8E6795B	31BF8F7D	C513746E
6074ACF6	94D857E5	7D81A1C3	892D5AD0	5B9EB69C	AF324D8F	466BBBA9	B2C740BA
D3F4D9C9	275822DA	CE01D4FC	3AAD2FEF	E81EC3A3	1CB238B0	F5EBCE96	01473585
A420ED1D	508C160E	B9D5E028	4D791B3B	9FCAF777	6B660C64	823FFA42	76930151
3C5CB061	C8F04B72	21A9BD54	D5054647	07B6AA0B	F31A5118	1A43A73E	EEEF5C2D
4B8884B5	BF247FA6	567D8980	A2D17293	70629EDF	84CE65CC	6D9793EA	993B68F9

This table contains 32-bit values in hexadecimal representation for each value (0 to 255) of the argument a in the function crctab32 [a]. The table should be used line-by-line in ascending order from top left (0) to bottom right (255). For instance, crctab32[10] is highlighted using a darker background and red colour.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

B.2 Use cases

B.2.1 Unidirectional communication

The most basic type of communication is unidirectional communication, where a safety application on one device (Controller A in Figure B.1) sends data to a safety application on another device (Controller B in Figure B.1).

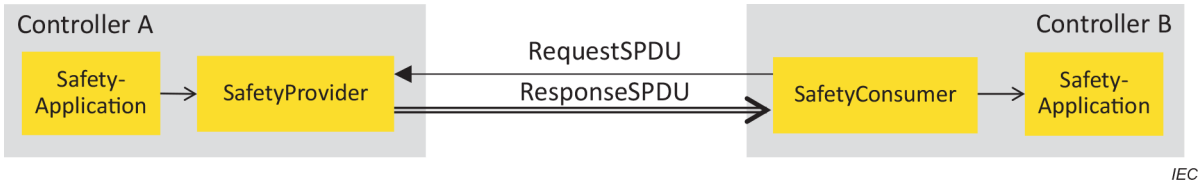


Figure B.1 – Unidirectional communication

This is accomplished by placing a *SafetyProvider* on Controller A and a *SafetyConsumer* on Controller B. The connection between *SafetyProvider* and *SafetyConsumer* can be established and terminated during runtime, allowing different consumers to connect to the same *SafetyProvider* at different times. Furthermore, the protocol is designed in such a way that it is necessary for the *SafetyConsumer* to know the parametrized set of IDs of the *SafetyProvider* such that it is able to safely check whether the received data is coming from the expected source. On the other hand, as *SafetyData* flows in one direction only, it is not necessary for the *SafetyProvider* to check the ID of the *SafetyConsumers*. Hence, Controller A can – one after another – serve an arbitrarily large number of *SafetyConsumers*, and new *SafetyConsumers* can be introduced into the system without having to update Controller A.

B.2.2 Bidirectional communication

Bidirectional communication means the exchange of data in both directions, which is accomplished by placing a *SafetyProvider* and a *SafetyConsumer* on each controller. Hence, bidirectional communication is realized using two safety connections according to this document. See Figure B.2 for an example.

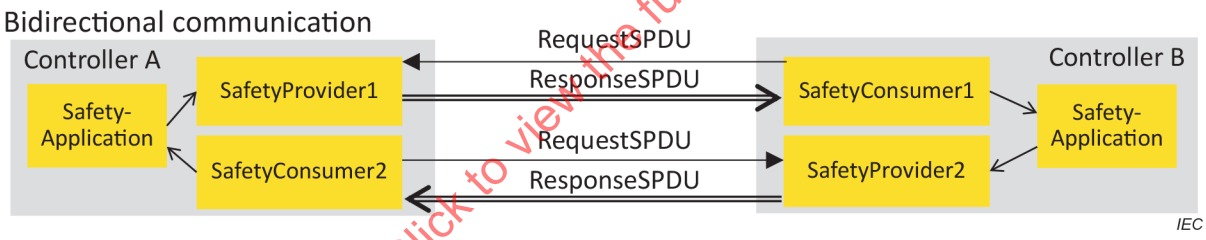


Figure B.2 – Bidirectional communication

NOTE Connections can be established and terminated during runtime.

B.2.3 Safety multicast

Multicast is defined as sending the same set of data from one device (Controller A) to several other devices (Controller B1, B2,…,BN) *simultaneously*.

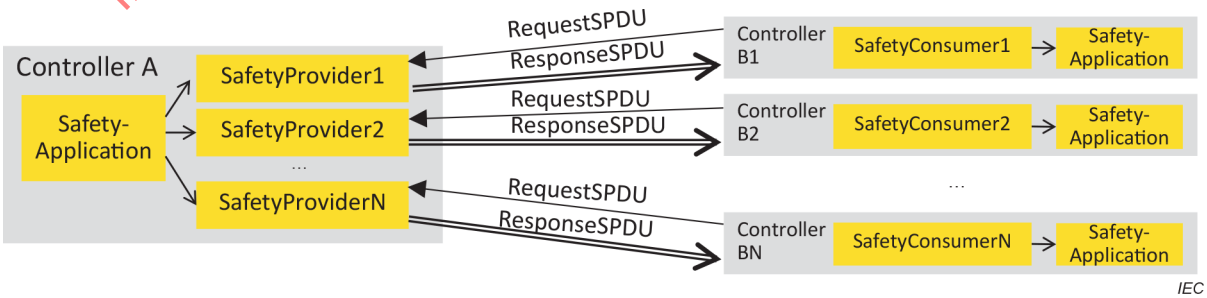


Figure B.3 – Safety multicast

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Safety multicast is accomplished by placing multiple *SafetyProviders* on Controller A, and one *SafetyConsumer* on each of the Controllers B1, B2, … BN. Each of the *SafetyProviders* running on Controller A is connecting to one of the *SafetyConsumers* running on one of the Controllers B1, B2, … BN. See Figure B.3 for an example.

The safety protocol in this document is designed in such a way that:

- the state machine of the *SafetyProvider* has a low number of states, and thus very low memory demands,
- all safety-related message-checks are executed on the consumer and thus the computational demand on the *SafetyProvider* is low.

Therefore, even if many *SafetyProviders* are instantiated on a device, the performance requirements will still be moderate.

The properties of simple unicast are also valid for safety multicast; different sets of consumers can connect to *SafetyProviders* at different times, and new *SafetyConsumers* can be introduced into the system without having to reconfigure the *SafetyProvider* instances. As all *SafetyProvider* instances send the same safety application data (the same data source), it is irrelevant from a safety point of view to which *SafetyProvider* instance a given *SafetyConsumer* is connected. Thus, all *SafetyProvider* instances can be parametrized with the same set of IDs for the *SafetyProvider*.

B.3 Use cases for operator acknowledgment

B.3.1 Explanation

This document supports operator acknowledgment both on the *SafetyProvider* side and on the *SafetyConsumer* side. For this purpose, both the interface of the *SafetyProvider* and the *SafetyConsumer* comprise a Boolean input called OperatorAckProvider and OperatorAckConsumer, respectively. The safety application can get the values of these parameters on the consumer side via the Boolean outputs OperatorAckRequested and OperatorAckProvider on the *SafetyConsumer*'s *SAPI* (see 6.3.4.2).

Subclauses B.3.2 to B.3.5 show some examples on how to use these inputs and outputs. Dashed lines indicate that the corresponding input or output is not used in the use case. For details, see 6.3.3 and 6.3.4.

B.3.2 Use case 1: unidirectional communication and OA on the SafetyConsumer side

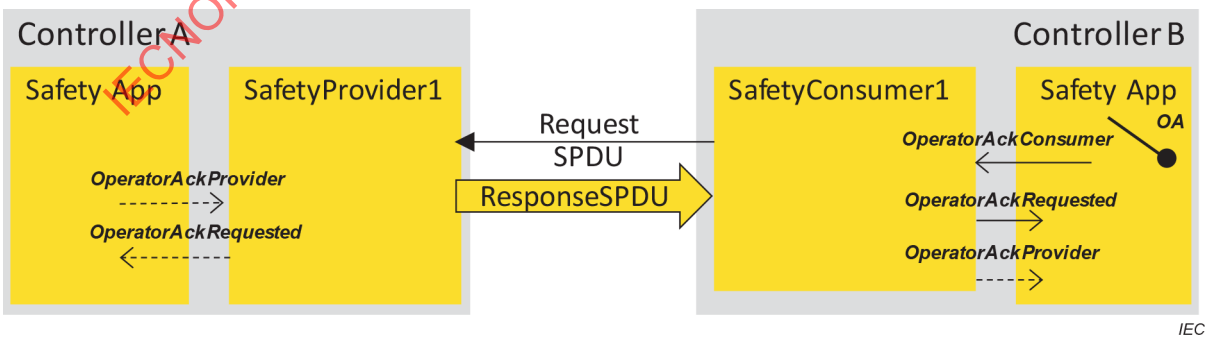


Figure B.4 – OA in unidirectional safety communication

In the scenario shown in Figure B.4, operator acknowledgment is done on the *SafetyConsumer* side, operator acknowledgment on the *SafetyProvider* side is not possible.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

B.3.3 Use case 2: bidirectional communication and dual OA

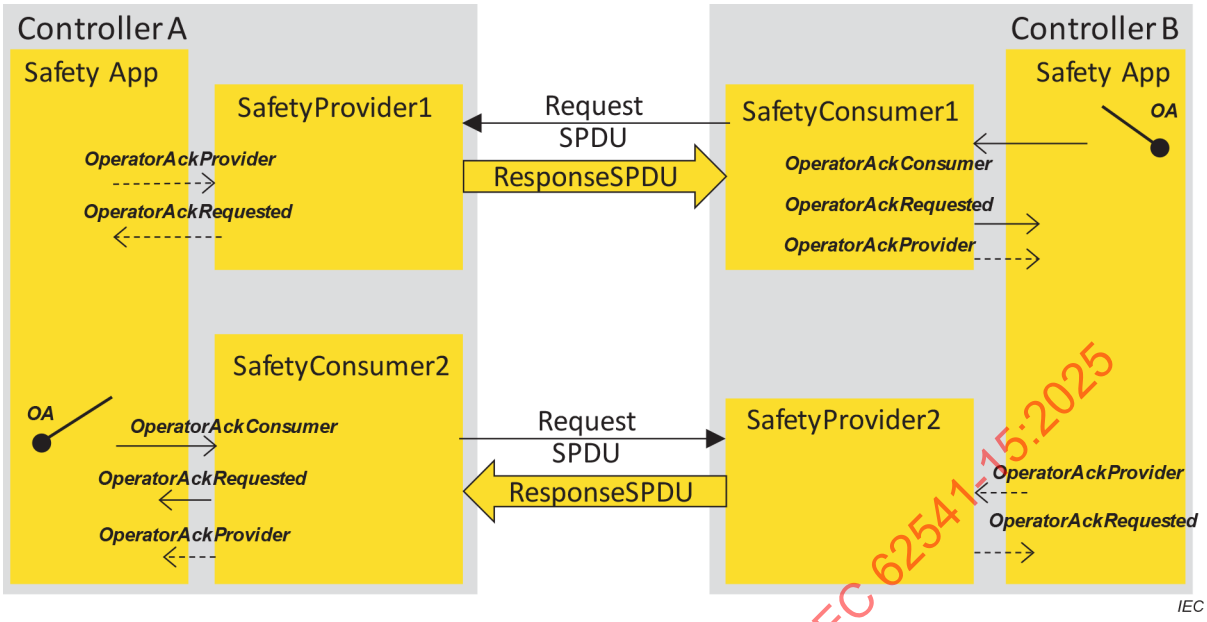


Figure B.5 – Two-sided OA in bidirectional safety communication

In the scenario shown in Figure B.5, operator acknowledgment is done independently for both directions.

B.3.4 Use case 3: bidirectional communication and single, one-sided OA

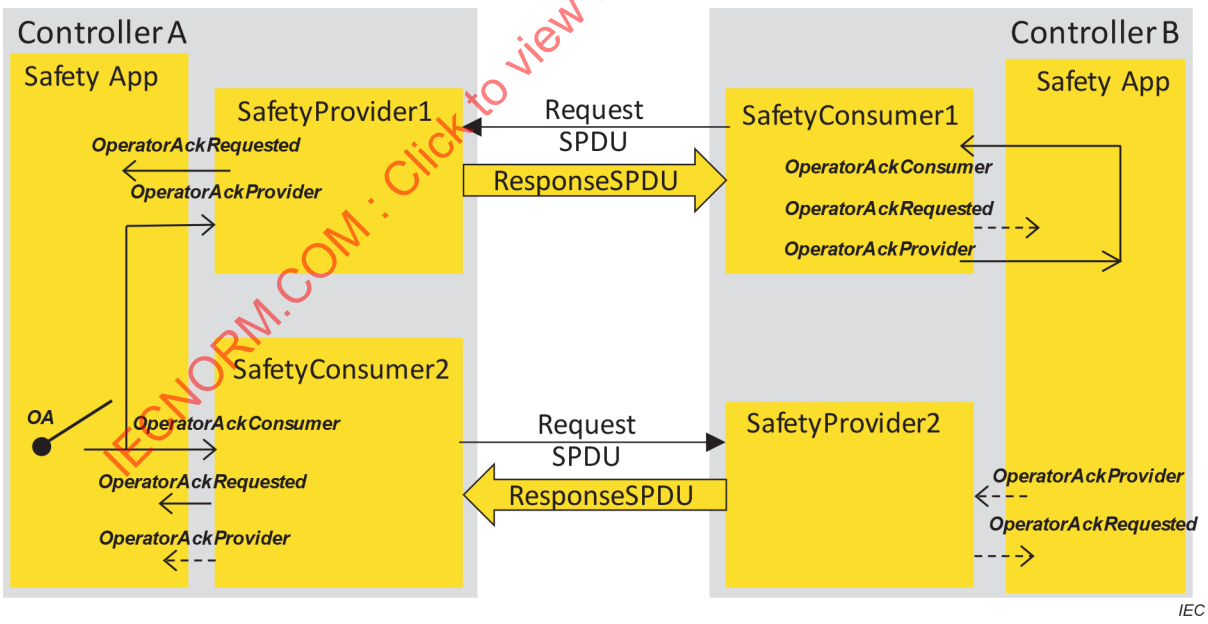


Figure B.6 – One sided OA in bidirectional safety communication

In the scenario of Figure B.6, an operator acknowledgment activated at controller A suffices for re-establishing the bidirectional connection. Both sides will cease delivering *fail-safe* values and continue sending *process values*. This is accomplished by connecting OperatorAckProvider with OperatorAckConsumer at the *SafetyConsumer* of controller B. Activating operator acknowledgment at controller B is not possible in this scenario.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

B.3.5 Use case 4: bidirectional communication and single, two-sided OA

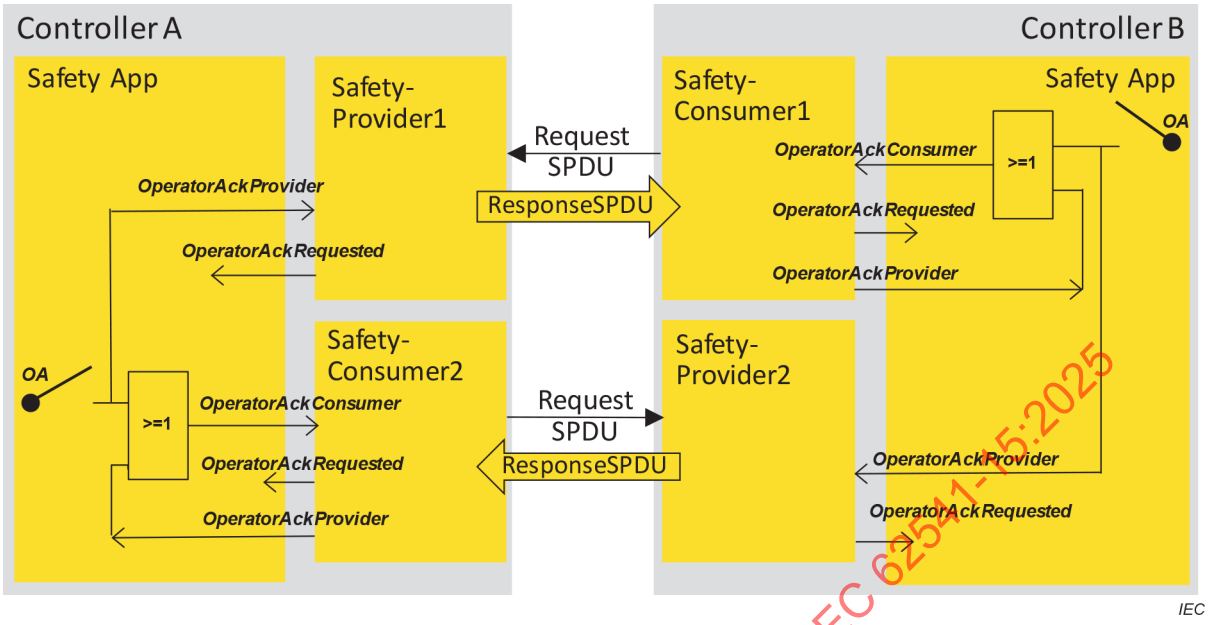


Figure B.7 – One sided OA on each side is possible

Figure B.7 shows a scenario where an operator acknowledgment activated at controller A or controller B suffices for re-establishing the bidirectional connection. Both sides will cease delivering *fail-safe* values and continue sending *process values*. This is accomplished by the logic circuits shown in the safety applications.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Annex C
(informative)

Information for assessment

This document does not make a statement on how to validate conformance. However, testing and validation of compliance can be subject to regulatory requirements.

Corresponding information regarding the testing and compliance with this document can be retrieved from the local National Committees of the IEC or from the relevant technology-specific organization for this document.

NOTE For this document, the relevant technology-specific organization is the OPC Foundation, see www.opcfoundation.com.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Bibliography

[1] Object Management Group, *Unified Modeling Language (UML)*, V2.5.1, 2017, https://www.omg.org/spec/UML/2.5.1/

[2] National Institute of Standards and Technology (NIST), *Computer Security Resource Center, Recommendation for Random Number Generation Using Deterministic Random Bit Generators*, SP 800-90A Rev. 1, June 2015

[3] Anwendungshinweise und Interpretationen (AIS) 20, *Functionality classes and evaluation methodology for physical random number generators*. Bundesamt für Sicherheit in der Informationstechnik (BSI), 1999

[4] Anwendungshinweise und Interpretationen (AIS) 31, *Functionality Classes and Evaluation Methodology for Physical Random Number Generators*, Bundesamt für Sicherheit in der Informationstechnik (BSI), 2001

[5] ISO/IEC 18031, *Information technology, Security techniques*. Random Bit Generation

[6] IEC 61000-6-7, *Electromagnetic compatibility (EMC) – Part 6-7: Generic standards – Immunity requirements for equipment intended to perform functions in a safety related system (functional safety) in industrial locations*

[7] IEC 61511 (all parts), *Functional safety – Safety instrumented systems for the process industry sector*

[8] IEC 62061, *Safety of machinery – Functional safety of safety-related control systems*

[9] ISO 13849 (all parts), *Safety of machinery – Safety-related parts of control systems*

[10] ISO 13849-1, *Safety of machinery – Safety-related parts of control systems – Part 1: General principles for design*

[11] IEC 62541-2, *OPC Unified Architecture – Part 2: Security*

[12] ISO 13849-2, *Safety of machinery – Safety-related parts of control systems – Part 2: Validation*

[13] IEC 62541-7, *OPC Unified Architecture – Part 7: Profiles*

[14] IEC 62541-8, *OPC Unified Architecture – Part 8: Data Access*

[15] OPC 10010 (all parts), *OPC Test Lab Specification*

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

SOMMAIRE

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

_____

ARCHITECTURE UNIFIÉE OPC –

Partie 15: Sécurité

AVANT-PROPOS

1) La Commission Électrotechnique Internationale (IEC) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de l'IEC). L'IEC a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. À cet effet, l'IEC – entre autres activités – publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de l'IEC"). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec l'IEC, participent également aux travaux. L'IEC collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
2) Les décisions ou accords officiels de l'IEC concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de l'IEC intéressés sont représentés dans chaque comité d'études.
3) Les Publications de l'IEC se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de l'IEC. Tous les efforts raisonnables sont entrepris afin que l'IEC s'assure de l'exactitude du contenu technique de ses publications; l'IEC ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de l'IEC s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de l'IEC dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de l'IEC et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
5) L'IEC elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de l'IEC. L'IEC n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
7) Aucune responsabilité ne doit être imputée à l'IEC, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de l'IEC, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de l'IEC ou de toute autre Publication de l'IEC, ou au crédit qui lui est accordé.
8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.
9) L'IEC attire l'attention sur le fait que la mise en application du présent document peut entraîner l'utilisation d'un ou de plusieurs brevets. L'IEC ne prend pas position quant à la preuve, à la validité et à l'applicabilité de tout droit de brevet revendiqué à cet égard. À la date de publication du présent document, l'IEC avait reçu notification qu'un ou plusieurs brevets pouvaient être nécessaires à sa mise en application. Toutefois, il y a lieu d'avertir les responsables de la mise en application du présent document que des informations plus récentes sont susceptibles de figurer dans la base de données de brevets, disponible à l'adresse https://patents.iec.ch. L'IEC ne saurait être tenue pour responsable de ne pas avoir identifié tout ou partie de tels droits de brevet.

L'IEC 62541-15 a été établie par le sous-comité 65C: Réseaux industriels, du comité d'études 65 de l'IEC: Mesure, commande et automation dans les processus industriels. Il s'agit d'une Norme internationale.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Le texte de cette Norme internationale est issu des documents suivants:

Projet	Rapport de vote
65C/1334/FDIS	65C/1339/RVD

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à son approbation.

La langue employée pour l'élaboration de cette Norme internationale est l'anglais.

Ce document a été rédigé selon les Directives ISO/IEC, Partie 2, il a été développé selon les Directives ISO/IEC, Partie 1 et les Directives ISO/IEC, Supplément IEC, disponibles sous www.iec.ch/members_experts/refdocs. Les principaux types de documents développés par l'IEC sont décrits plus en détail sous www.iec.ch/publications.

Tout au long du présent document et des autres parties référencées de la série IEC 62541, certaines conventions documentaires sont utilisées:

Le format *italique* est utilisé pour mettre en évidence un terme défini ou une définition qui apparaît à l'Article 3 dans l'une des parties de la série.

Le format *italique* est également utilisé pour mettre en évidence le nom d'un paramètre d'entrée ou de sortie de service, ou le nom d'une structure ou d'un élément de structure habituellement défini dans les tableaux.

Par ailleurs, les *termes* et *noms en italique* sont, à quelques exceptions près, écrits en camel-case (pratique qui consiste à joindre, sans espace, les éléments des mots ou expressions composés, la première lettre de chaque élément étant en majuscule). Par exemple, le terme défini est *AddressSpace* et non Espace d'Adressage. Cela permet de mieux comprendre qu'il existe une définition unique pour *AddressSpace*, et non deux définitions distinctes pour Espace et pour Adressage.

Une liste de toutes les parties de la série IEC 62541, publiées sous le titre général *Architecture unifiée OPC*, se trouve sur le site web de l'IEC.

Le comité a décidé que le contenu de ce document ne sera pas modifié avant la date de stabilité indiquée sur le site web de l'IEC sous webstore.iec.ch dans les données relatives au document recherché. À cette date, le document sera

- reconduit,
- supprimé, ou
- révisé.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

INTRODUCTION

La sécurité OPC UA étend l'OPC UA pour satisfaire aux exigences de sécurité fonctionnelle définies dans les séries de normes IEC 61508 et IEC 61784-3.

La Figure 1 représente la relation entre le présent document et les normes de sécurité et OPC UA pertinentes dans un environnement industriel. Une flèche du Document A au Document B signifie que le Document A est référencé dans le Document B. Cette référence peut être normative ou informative. Toutes ces normes ne sont pas applicables ou exigées pour un produit donné.

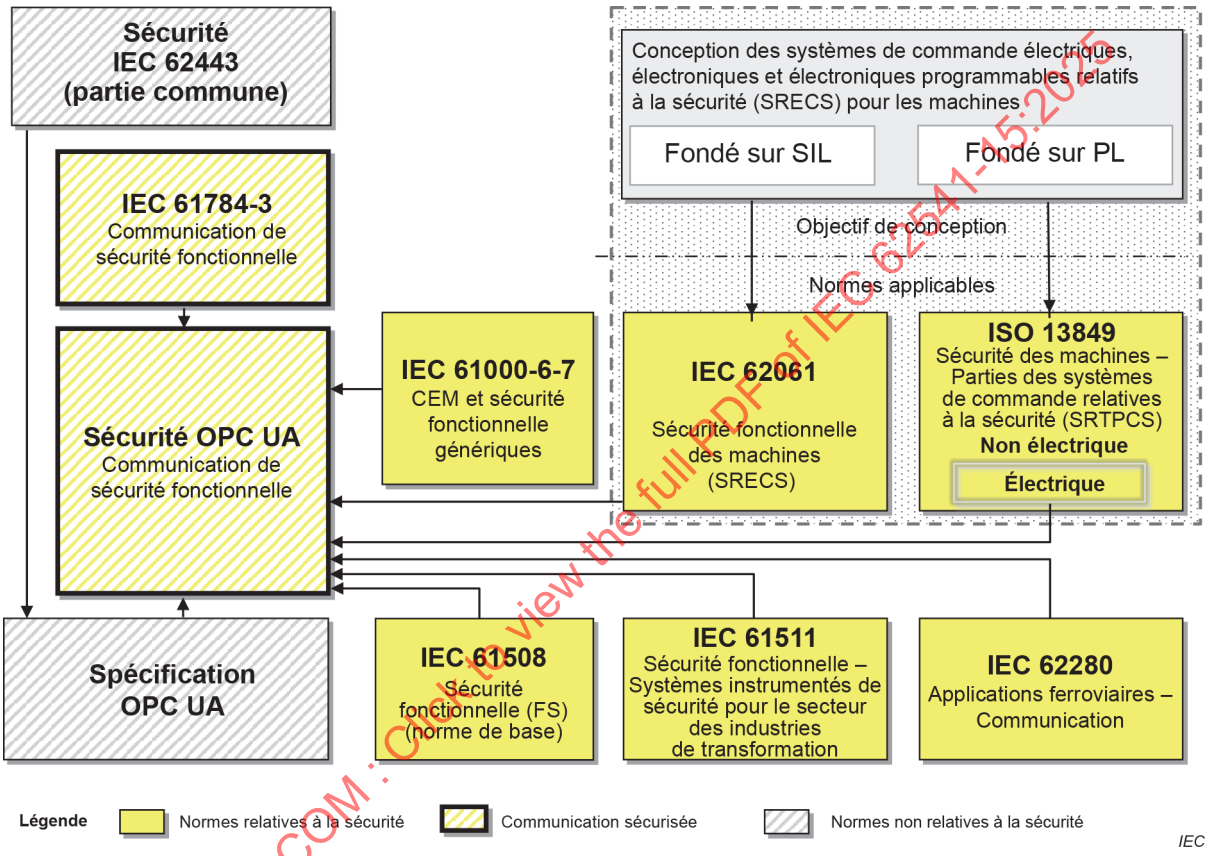


Figure 1 – Relations entre la sécurité OPC UA et d'autres normes

La mise en œuvre du présent document permet de détecter tous les types d'erreurs de communication rencontrés dans les couches réseau inférieures. Si une erreur est détectée, ces informations sont partagées avec les applications de sécurité de la couche utilisateur, qui peuvent alors agir de manière appropriée, par exemple en basculant dans un état de sécurité.

Le document décrit le comportement des points d'extrémité individuels pour une communication sûre, ainsi que le *Modèle d'information* OPC UA qui est utilisé pour accéder à ces points d'extrémité.

Le présent document est indépendant de l'application et n'impose aucune exigence quant à la structure et à la longueur des données d'application. Il est attendu que les exigences spécifiques à l'application soient décrites dans les spécifications associées appropriées.

Le présent document peut être utilisé pour des applications qui exigent une sécurité fonctionnelle jusqu'au *niveau d'intégrité de sécurité* (*SIL*) 4.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

ARCHITECTURE UNIFIÉE OPC –

Partie 15: Sécurité

1 Domaine d'application

Le présent document décrit une *couche de communication de sécurité* (services et protocole) pour l'échange de *SafetyData* à l'aide des mécanismes de l'IEC 62541. Il identifie les principes qui s'appliquent aux communications de sécurité fonctionnelle définies dans l'IEC 61784-3, associés à cette *couche de communication de sécurité*. Cette *couche de communication de sécurité* est destinée à être mise en œuvre sur les appareils de *sécurité* uniquement.

NOTE 1 Le présent document cible la communication de contrôleur à contrôleur. Cependant, la facilité d'extension à d'autres cas d'utilisation (par exemple, communication au niveau du terrain OPC UA) a déjà été prise en compte dans la conception du présent document.

NOTE 2 Le présent document ne traite pas des aspects relatifs à la sécurité électrique et à la sécurité intrinsèque. La sécurité électrique concerne les dangers comme les chocs électriques. La sécurité intrinsèque concerne les dangers associés aux atmosphères explosibles.

Le présent document définit les mécanismes de transmission des messages relatifs à la sécurité entre les participants d'un réseau, en utilisant la technologie OPC UA conformément aux exigences de la série IEC 61508 et de l'IEC 61784-3 concernant la sécurité fonctionnelle. Ces mécanismes peuvent être utilisés dans différentes applications industrielles, par exemple la commande de processus, la fabrication, l'automatisation et les machines.

Le présent document fournit des lignes directrices aux développeurs, ainsi qu'aux évaluateurs d'appareils et de systèmes conformes.

NOTE 3 Le *SIL* ainsi revendiqué pour un système dépend de la mise en œuvre du présent document au sein du système (la mise en œuvre du présent document dans un appareil normal ne suffit pas à le qualifier d'appareil de sécurité).

2 Références normatives

Les documents suivants sont cités dans le texte de sorte qu'ils constituent, pour tout ou partie de leur contenu, des exigences du présent document. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

IEC 61508 (toutes les parties), *Sécurité fonctionnelle des systèmes électriques/électroniques/électroniques programmables relatifs à la sécurité.*

IEC 61784-3:2021, *Réseaux de communication industriels – Profils – Partie 3: Bus de terrain de sécurité fonctionnelle – Règles générales et définitions de profils.*

IEC 62443 (toutes les parties), *Sécurité des systèmes d'automatisation et de commande industrielles.*

IEC 62541-1:2020, *Architecture unifiée OPC – Partie 1: Vue d'ensemble et concepts.*

IEC 62541-3:2020, *Architecture unifiée OPC – Partie 3: Modèle d'espace d'adressage.*

IEC 62541-4:2020, *Architecture unifiée OPC – Partie 4: Services.*

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

IEC 62541-5:2020, *Architecture unifiée OPC – Partie 5: Modèle d'information.*

IEC 62541-6:2020, *Architecture unifiée OPC – Partie 6: Mappings.*

IEC 62541-14, *Architecture unifiée OPC – Partie 14: PubSub.*

ISO/IEC 9834-8:2014, *Technologies de l'information – Procédures opérationnelles pour les organismes d'enregistrement d'identificateur d'objet – Partie 8: Génération des identificateurs uniques universels (UUID) et utilisation de ces identificateurs dans les composants d'identificateurs d'objets.*

3 Termes, définitions, symboles, abréviations et conventions

3.1 Termes et définitions

Pour les besoins du présent document, les termes et définitions de l'IEC 62541-1:2020, l'IEC 62541-3:2020, l'IEC 62541-4:2020, l'IEC 62541-6:2020 ainsi que les suivants s'appliquent.

L'ISO et l'IEC tiennent à jour des bases de données terminologiques destinées à être utilisées en normalisation, consultables aux adresses suivantes:

- IEC Electropedia: disponible à l'adresse https://www.electropedia.org/
- ISO Online browsing platform: disponible à l'adresse https://www.iso.org/obp

NOTE Des concepts de modélisation de l'information de l'IEC 62541 sont utilisés pour décrire les concepts du présent document.

3.1.1 Termes et définitions communs

3.1.1.1
contrôle de redondance cyclique
CRC
<valeur> donnée redondante déduite, et enregistrée ou transmise simultanément, d'un bloc de données afin de détecter toute corruption des données

<méthode> procédure utilisée pour calculer les données redondantes

Note 1 à l'article: Les termes "code CRC" et "signature CRC", ainsi que les étiquettes comme CRC1, CRC2, peuvent également être utilisés dans le présent document pour se référer aux données redondantes.

[SOURCE: IEC 61784-3:2021, 3.10]

3.1.1.2
erreur
écart ou discordance entre une valeur ou une condition calculée, observée ou mesurée et la valeur ou la condition vraie, prescrite ou théoriquement correcte

Note 1 à l'article: Les erreurs peuvent être causées par des erreurs de conception du matériel/logiciel et/ou des informations altérées du fait d'un brouillage électromagnétique et/ou autres effets.

Note 2 à l'article: Les erreurs ne produisent pas nécessairement une défaillance ou une anomalie.

[SOURCE: IEC 60050-192:2024, 192-03-02, modifié – notes ajoutées]

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

3.1.1.3
défaillance
cessation de l'aptitude d'une unité fonctionnelle à accomplir une fonction requise ou à fonctionner comme prévu

Note 1 à l'article: Une défaillance peut être causée par une erreur (problème de conception matérielle/logicielle ou rupture de message, par exemple).

[SOURCE: IEC 61508-4:2010, 3.6.4, modifié – notes et figures supprimées, nouvelle note à l'article ajoutée]

3.1.1.4
anomalie
condition anormale qui peut entraîner une réduction de capacité ou la perte de capacité d'une unité fonctionnelle à accomplir une fonction requise

Note 1 à l'article: L'IEV 191-05-01 définit le terme "fault" (en français "panne") comme un état d'inaptitude à accomplir une fonction requise, en excluant l'inaptitude due à la maintenance préventive, à d'autres actions programmées ou à un manque de ressources extérieures.

[SOURCE: IEC 61508-4:2010, 3.6.1, modifié – référence à la figure supprimée]

message
<théorie de l'information et théorie des communications> suite ordonnée de caractères (généralement des octets) destinée à communiquer des informations

[SOURCE: ISO/IEC 2382:2015, 2123031, modifié – insertion de "(généralement des octets)", suppression des notes et de la source]

3.1.1.5
niveau de performance
PL
niveau discret d'aptitude de parties relatives à la sécurité à réaliser une fonction de sécurité dans des conditions prévisibles

Note 1 à l'article: L'abréviation "PL" est dérivée du terme anglais développé correspondant "performance level".

[SOURCE: ISO 13849-1:2023, 3.1.5]

3.1.1.6
probabilité d'erreurs résiduelles
probabilité de non-détection d'une erreur par les mesures de sécurité *SCL*

[SOURCE: IEC 61784-3:2021, 3.1.35]

3.1.1.7
taux d'erreurs résiduelles
taux statistique de défaut de détection d'erreurs par les mesures de sécurité *SCL*

[SOURCE: IEC 61784-3:2021, 3.1.36]

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

3.1.1.8
couche de communication de sécurité
SCL
couche de communication située au-dessus de la pile de communication IEC 62541 qui comprend toutes les mesures supplémentaires nécessaires qui permettent d'assurer la transmission de données en toute sécurité conformément aux exigences de l'IEC 61508

Note 1 à l'article: La *SCL* fournit plusieurs services, les plus importants étant les services *SafetyProvider* et *SafetyConsumer*.

Note 2 à l'article: L'abréviation "SCL" est dérivée du terme anglais développé correspondant "safety communication layer".

[SOURCE: IEC 61784-3:2021, 3.1.39, modifié – "FAL" remplacé par "pile de communication IEC 62541", note à l'article ajoutée]

3.1.1.9
temps de réponse de la fonction de sécurité
temps écoulé dans le cas le plus défavorable à la suite de l'activation d'un capteur de sécurité relié à un bus de terrain, avant que ne soit atteint l'état de sécurité correspondant de ses actionneurs de sécurité, du fait d'erreurs ou de défaillances dans la fonction de sécurité

Note 1 à l'article: Ce concept, adopté dans l'IEC 61784-3:2021, 5.2.4, est traité par les profils de communication de sécurité fonctionnelle définis dans la série de documents IEC 61784-3.

[SOURCE: IEC 61784-3:2021, 3.1.44]

3.1.1.10
niveau d'intégrité de sécurité
SIL
niveau discret (parmi quatre possibles) correspondant à une gamme de valeurs d'intégrité de sécurité, où le niveau 4 d'intégrité de sécurité possède le plus haut degré d'intégrité et le niveau 1 possède le plus bas

Note 1 à l'article: Les objectifs chiffrés de défaillance (voir l'IEC 61508-4:2010, 3.5.17) pour les quatre *niveaux d'intégrité de sécurité* sont indiqués dans le Tableau 2 et le Tableau 3 de l'IEC 61508-1:2010.

Note 2 à l'article: Les *niveaux d'intégrité de sécurité* sont utilisés pour spécifier les exigences concernant l'intégrité de sécurité des fonctions de sécurité à allouer aux systèmes E/E/PE relatifs à la sécurité.

Note 3 à l'article: Un *niveau d'intégrité de sécurité* (*SIL*) ne constitue pas une propriété d'un système, sous-système, élément ou composant. L'interprétation correcte de l'expression "système relatif à la sécurité à *SIL* n" (où n est 1, 2, 3 ou 4) signifie que le système est potentiellement capable de prendre en charge les fonctions de sécurité avec un *niveau d'intégrité de sécurité* jusqu'à n.

Note 4 à l'article: L'abréviation "SIL" est dérivée du terme anglais développé correspondant "safety integrity level".

[SOURCE: IEC 61508-4:2010, 3.5.8]

3.1.1.11
mesure de sécurité
mesure permettant de contrôler les erreurs de communication éventuelles, qui est conçue et mise en œuvre conformément aux exigences de l'IEC 61508

Note 1 à l'article: Dans la pratique, plusieurs mesures de sécurité sont combinées pour atteindre le *niveau d'intégrité de sécurité* exigé.

Note 2 à l'article: Les erreurs de communication et les mesures de sécurité associées sont décrites dans l'IEC 61784-3:2021, 5.3 et 5.4.

[SOURCE: IEC 61784-3:2021, 3.1.46]

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

3.1.1.12
PDU de sécurité
SPDU
PDU transféré par le *canal de communication de sécurité*

Note 1 à l'article: La SPDU peut comporter plusieurs exemplaires des *SafetyData* qui utilisent des structures de codage et des fonctions de hachage différentes, associées à des parties explicites de protections supplémentaires, par exemple une clé, un nombre de séquences ou un mécanisme d'horodatage.

Note 2 à l'article: Les *SCL* redondantes peuvent fournir deux versions différentes de la SPDU en vue de son insertion dans des champs séparés de la trame IEC 62541.

Note 3 à l'article: L'abréviation "SPDU" est dérivée du terme anglais développé correspondant "safety protocol data unit".

[SOURCE: IEC 61784-3:2021, 3.1.47]

3.1.2 Termes et définitions supplémentaires

3.1.2.1
à sécurité intrinsèque (Failsafe)
aptitude d'un système qui, par des mesures techniques ou organisationnelles pertinentes, protège contre les dangers de manière déterministe ou en réduisant le risque à un niveau tolérable

Note 1 à l'article: Équivalent à sécurité fonctionnelle.

3.1.2.2
valeurs de substitution Failsafe
valeurs de substitution à sécurité intrinsèque
FSV
valeurs qui sont émises ou fournies en remplacement des valeurs de processus lorsque la fonction de sécurité est définie sur un état de sécurité intrinsèque

Note 1 à l'article: Dans le présent document, les valeurs de substitution Failsafe (FSV) sont toujours définies sur la valeur binaire "0".

Note 2 à l'article: L'abréviation "FSV" est dérivée du terme anglais développé correspondant "fail-safe substitute values".

3.1.2.3
fanion
valeur à un bit utilisée pour indiquer certaines informations de statut ou de contrôle

3.1.2.4
identificateur global unique
GUID
nombre de 128 bits utilisé pour identifier des informations dans des systèmes informatiques

Note 1 à l'article: Le terme "identificateur unique universel" (UUID) est également utilisé.

Note 2 à l'article: Dans le présent document, la version 4 de l'UUID est utilisée.

Note 3 à l'article: L'abréviation "GUID" est dérivée du terme anglais développé correspondant "globally unique identifier".

3.1.2.5
MonitoringNumber
MNR
moyen utilisé pour assurer l'ordre correct parmi les PDU de sécurité transmises et pour surveiller le retard de communication

Note 1 à l'article: Instance de nombre de séquences décrite dans l'IEC 61784-3.

Note 2 à l'article: Le *MNR* commence à une valeur aléatoire et est incrémenté à chaque demande. Il passe à une valeur de seuil minimale différente de zéro.

Note 3 à l'article: Le *MNR* transmis est protégé par la signature *CRC* transmise de la ResponseSPDU.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

3.1.2.6
non relatif à la sécurité
prédicat signifiant que l'objet correspondant est un objet "normalisé" et qu'il n'a pas été conçu et mis en œuvre pour satisfaire aux exigences relatives à la sécurité fonctionnelle

3.1.2.7
mappeur OPC UA
partie de la mise en œuvre du présent document non relative à la sécurité, qui associe par mapping la SPDU avec les services réels de l'IEC 62541

Note 1 à l'article: Selon les services de l'IEC 62541 qui sont utilisés (par exemple, *Client/Serveur* ou *PubSub*), différents mappeurs peuvent être spécifiés.

3.1.2.8
valeurs de processus
PV
données d'entrée et de sortie (d'une PDU de sécurité) nécessaires au contrôle d'un processus automatisé

Note 1 à l'article: L'abréviation "PV" est dérivée du terme anglais développé correspondant "process values".

3.1.2.9
qualificatif
attribut (binaire ou booléen) indiquant si la valeur correspondante est valide ou non (par exemple en tant que valeur de substitution FailSafe)

3.1.2.10
SafetyAutomationComponent
SafetyAC
partenaire de communication dans une liaison de sécurité unidirectionnelle

Note 1 à l'article: Un *SafetyAutomationComponent* peut être un *SafetyProvider* (source de données), et/ou un *SafetyConsumer* (collecteur de données).

3.1.2.11
SafetyConsumer
entité (en général logicielle) qui met en œuvre le collecteur de données d'une liaison de sécurité unidirectionnelle

3.1.2.12
SafetyData
données d'application transmises par un réseau de sécurité qui utilise un protocole de sécurité

Note 1 à l'article: La *couche de communication de sécurité* n'assure pas la sécurité des données proprement dites, mais uniquement la transmission en toute sécurité de ces dernières.

3.1.2.13
SafetyProvider
entité (en général logicielle) qui met en œuvre la source de données d'une liaison de sécurité unidirectionnelle

3.1.2.14
SafetyBaseID
ID d'authenticité généré de manière aléatoire, qui est utilisé pour authentifier en toute sécurité les *SafetyProviders* ayant le même *SafetyProviderID*

Note 1 à l'article: Avec le *SafetyProviderID*, il s'agit d'une instance d'authentification de connexion, comme cela est décrit dans l'IEC 61784-3.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

3.1.2.15
SafetyProviderID
identificateur local unique affecté par l'utilisateur qui est utilisé pour authentifier en toute sécurité les *SafetyProviders* dans une certaine zone

Note 1 à l'article: Avec le *SafetyBaseID*, il s'agit d'une instance d'authentification de connexion, comme cela est décrit dans l'IEC 61784-3.

Note 2 à l'article: Tous les *SafetyProviders* dans une zone ainsi définie peuvent partager un *SafetyBaseID* identique.

3.1.2.16
système de transmission normalisé
partie du système de transmission (mise en œuvre au niveau du matériel et du logiciel) qui n'est pas mise en œuvre conformément aux normes de sécurité

Note 1 à l'article: Le présent document utilise les services du *système de transmission normalisé* pour transmettre des paquets de sécurité préconçus.

3.2 Symboles et abréviations

Pour les besoins du présent document, les symboles et abréviations suivants s'appliquent.

3.2.1 Abréviations de l'IEC 61784-3

CRC	contrôle de redondance cyclique	
PDU (Protocol Data Unit)	unité de données de protocole	[ISO/IEC 7498-1]
PL (Performance Level)	niveau de performance	[ISO 13849-1]
PLC (Programmable Logic Controller)	automate programmable	
SCL (Safety Communication Layer)	couche de communication de sécurité	
SIL (Safety Integrity Level)	niveau d'intégrité de sécurité	[IEC 61508-4]
SPDU (Safety Protocol Data Unit)	PDU de sécurité, unité de données de protocole de sécurité	

3.2.2 Symboles et abréviations supplémentaires

3.2.2.1 Abréviations

FSV (Fail-safe Substitute Values)	valeurs de substitution Failsafe valeurs de substitution à sécurité intrinsèque
IHM	interface homme/machine
ID	identificateur
LSB (Least Significant Bit)	bit de poids faible
MNR	MonitoringNumber
MSB (Most Significant Bit)	bit de poids fort
OA (Operator Acknowledgment)	acquittement de l'opérateur
OPC UA PI (OPC UA Platform Interface)	interface de plateforme OPC UA
PI (Platform Interface)	interface de plateforme
PV (Process Values)	valeurs de processus
SAPI (Safety Application Program Interface)	interface du programme d'application de sécurité
SFRT (Safety Function Response Time)	temps de réponse de la fonction de sécurité
SPI (Safety Parameter Interface)	interface de paramètres de sécurité
STrailer (Safety Trailer)	queue de sécurité
TRA (Threat and Risk Analysis)	analyse des menaces et des risques

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

3.2.2.2 Symboles

p	probabilité d'erreurs sur les éléments binaires
$P_{re,cond}$	probabilité d'erreurs résiduelles conditionnelles

3.3 Conventions

3.3.1 Conventions générales

Des traits de soulignement sont utilisés pour séparer les termes ou les noms dans lesquels deux lettres majuscules correspondant à des abréviations se suivent ou sont utilisées pour séparer un suffixe.

L'abréviation "F" se rapporte aux éléments, technologies, systèmes et unités *relatifs à la sécurité* (*à sécurité intrinsèque*, sécurité fonctionnelle).

Les données par défaut qui sont utilisées en cas de défaillance d'une unité ou d'erreurs sont appelées *valeurs de substitution Failsafe* (*FSV*) et sont définies sur la valeur binaire "0".

Les bits réservés ("res") sont définis sur "0" et sont ignorés par le récepteur afin d'éviter les problèmes pour les futures versions du présent document.

La notation 0x... représente une valeur hexadécimale.

3.3.2 Conventions pour la numérotation des exigences

Les exigences du présent document sont désignées sous la forme *[RQx.yz]*, où *x* indique le numéro de l'article, *y* est un compteur et *z* est un caractère facultatif pour relier des exigences étroitement liées. Les exemples de désignations d'exigences suivants sont valides: [RQ8.15] (exigence 15 de l'Article 8); [RQ47.11a], [RQ47.11b] (exigences 11a et 11b de l'Article 47, qui sont étroitement liées).

La numérotation initiale des exigences a été choisie de sorte que les compteurs de chaque article soient donnés dans l'ordre croissant. Toutefois, l'ajout d'exigences supplémentaires introduit des écarts par rapport à cette règle, car les exigences existantes doivent conserver leur désignation initiale.

Pour obtenir un index informatif de toutes les exigences du présent document, voir le 10.3.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

3.3.3 Conventions dans les diagrammes d'états

Voir le Tableau 1 pour les conventions utilisées dans les diagrammes d'états.

Tableau 1 – Conventions utilisées dans les diagrammes d'états

Convention	Signification
:=	Affectation: la valeur d'un élément à gauche est remplacée par la valeur de l'élément à droite.
<	Inférieur à: condition logique donnant la valeur TRUE si et seulement si un élément à gauche est inférieur à l'élément à droite.
<=	Inférieur ou égal à: condition logique donnant la valeur TRUE si et seulement si un élément à gauche est inférieur ou égal à l'élément à droite.
>	Supérieur à: condition logique donnant la valeur TRUE si et seulement si l'élément à gauche est supérieur à l'élément à droite.
>=	Supérieur ou égal à: condition logique donnant la valeur TRUE si et seulement si l'élément à gauche est supérieur ou égal à l'élément à droite.
==	Égalité: condition logique donnant la valeur TRUE si et seulement si l'élément à gauche est égal à l'élément à droite.
<>	Inégalité: condition logique donnant la valeur TRUE si et seulement si l'élément à gauche est différent de l'élément à droite.
&&	"AND" logique (opération sur des résultats ou des valeurs binaires).
||	"OR" logique (opération sur des résultats ou des valeurs binaires).
⊕	"XOR" logique (opération sur des valeurs binaires ou numériques).
[..]	Conditions de garde UML: la transition correspondante est activée si et seulement si la garde a la valeur TRUE.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

4 Vue d'ensemble de la sécurité OPC UA

4.1 Généralités

Le présent document spécifie une *couche de communication de sécurité* (*SCL*) qui permet à des appareils relatifs à la sécurité d'utiliser les services de l'IEC 62541 pour l'échange sécurisé de données relatives à la sécurité. Un appareil de sécurité qui met correctement en œuvre la sécurité OPC UA peut échanger des données relatives à la sécurité et satisfaire ainsi aux exigences de la série IEC 61508 et de l'IEC 61784-3. Le présent document utilise un *MonitoringNumber*, une temporisation, un ensemble d'ID et un *contrôle de redondance cyclique* (*CRC*) pour détecter toutes les erreurs de communication possibles qui peuvent se produire dans le *système de transmission normalisé* IEC 62541 sous-jacent. Ces *mesures de sécurité* ont fait l'objet d'une évaluation quantitative et offrent une probabilité de défaillance dangereuse par heure (PFH, *Probability of dangerous Failure per Hour*) et une probabilité de défaillance dangereuse sur sollicitation (PFD, *Probability of dangerous Failure on Demand*) suffisantes pour développer des applications *relatives à la sécurité* avec un *niveau d'intégrité de sécurité* allant jusqu'à *SIL*4.

La sécurité OPC UA est en soi une solution générale indépendante de l'application. La longueur et la structure des données envoyées sont définies par l'application de sécurité. Il est cependant prévu que les spécifications associées dépendant de l'application (par exemple, les équipements de protection électrosensibles, les entraînements électriques avec fonctions de sécurité, les presses de formage, la sécurité des robots et les véhicules guidés automatisés) soient définies par des experts de l'application dans les spécifications appropriées qui sont associées à l'OPC UA.

4.2 Aspects relatifs à la mise en œuvre

[RQ4.1] Toutes les mesures techniques de détection d'erreurs décrites dans le présent document doivent être mises en œuvre dans la *SCL* des appareils conçus conformément à la série IEC 61508 et doivent satisfaire au *SIL* cible.

4.3 Caractéristiques

- Exécuté sur:
  - le *Client/Serveur* IEC 62541 avec le *Jeu de services Method*;
  - le *PubSub* IEC 62541.
- Du point de vue de l'architecture: facilité d'extension à d'autres modes de communication.
- Objectif: aucune modification du cadre OPC UA existant.
- Les diagrammes d'états du présent document sont indépendants du *mappeur OPC UA*, ce qui permet un échange simplifié du mappeur.
- Compatible avec les canaux de transmission sans fil.
- Exigences peu contraignantes concernant les nœuds du réseau de sécurité:
  - aucune synchronisation d'horloge n'est nécessaire (aucune exigence concernant la précision entre les horloges sur différents nœuds);
  - dans le *SafetyConsumer*, un temporisateur local relatif à la sécurité est exigé pour la mise en œuvre du *SafetyConsumerTimeout*. La précision de ce temporisateur dépend des exigences de temporisation de l'application de sécurité.
- Sécurité de bout en bout: les *SafetyData* fonctionnelles sont transportées entre deux terminaux de sécurité sur un réseau normalisé qui n'est pas conforme sur le plan de la sécurité fonctionnelle. Cela comprend les couches de transport inférieures telles que la pile de l'IEC 62541, les supports physiques sous-jacents et les éléments de réseau *non relatifs à la sécurité* (par exemple, routeurs et commutateurs).
- Systèmes "dynamiques":
  - les partenaires de communication de sécurité peuvent changer pendant l'exécution;
  - le nombre de partenaires de communication de sécurité peut augmenter ou diminuer, ou les deux.
- Des chaînes de texte bien définies sont utilisées à des fins de diagnostic.
- La communication de sécurité et la communication normalisée sont indépendantes. Toutefois, les appareils normalisés et les appareils de sécurité peuvent utiliser simultanément le même *système de transmission normalisé*.
- La sécurité fonctionnelle peut être obtenue sans utiliser de *systèmes de transmission normalisés* structurellement redondants, c'est-à-dire qu'une approche monocanal peut être utilisée. La redondance peut être éventuellement utilisée pour améliorer la disponibilité.
- À des fins de diagnostic, la dernière *SPDU* envoyée et reçue est accessible dans le *Modèle d'information* du *SafetyProvider*.
- Longueur des données utilisateur: entre 1 octet et 1 500 octets, structures de *DataTypes* de base (voir 6.2.5).

4.4 Politique de sûreté

Dans l'application finale, un environnement de sûreté approprié est nécessaire pour protéger à la fois l'environnement opérationnel et les systèmes relatifs à la sécurité.

Le présent document ne traite pas des aspects relatifs à la sûreté et ne prévoit aucune exigence en matière de sûreté.

Une analyse des menaces et des risques (TRA) conforme à la série IEC 62443 doit être effectuée au niveau du système d'application finale.

Lors des essais de conformité au présent document, les aspects relatifs à la sûreté ne font pas partie du domaine d'application, car il est admis par hypothèse que les mécanismes de base sous-jacents (c'est-à-dire les *Méthodes*) fournissent déjà une sûreté adéquate.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

5 Généralités

5.1 Documents externes de spécifications applicables au profil

Aucun autre document externe ne fournit de spécifications en plus des Références normatives données à l'Article 2.

5.2 Exigences fonctionnelles de sécurité

Les exigences suivantes s'appliquent à l'élaboration du présent document:

a) communication de sécurité conforme à un *niveau d'intégrité de sécurité* allant jusqu'à *SIL*4 (voir la série IEC 61508) et à un niveau PL e (voir l'ISO 13849-1);

b) combinaison d'appareils *SIL*1 à 4 conformes au présent document et d'appareils *non relatifs à la sécurité* sur un même réseau de communication;

c) la mise en œuvre du protocole de transmission de sécurité est limitée à la couche de sécurité;

d) la surveillance de temporisation relative à la sécurité est mise en œuvre dans la couche de sécurité;

e) la communication de sécurité satisfait aux exigences de l'IEC 61784-3;

f) [RQ5.1] le présent document est destiné à être mis en œuvre exclusivement dans les appareils de sécurité. Les exceptions (par exemple, pour le débogage, la simulation, les essais et la mise en service) doivent faire l'objet d'une discussion avec un organisme notifié.

5.3 Mesures de sécurité

[RQ5.2] Pour la mise en œuvre du présent document, les *mesures de sécurité* suivantes doivent être appliquées: *MonitoringNumber*; temporisation avec réception dans le *SafetyConsumer*; ensemble d'ID pour le *SafetyProvider*; contrôle d'intégrité des données.

Ensemble, ces *mesures de sécurité* traitent toutes les erreurs de transmission possibles énumérées dans l'IEC 61784-3:2021, 5.5 (voir Tableau 2).

[RQ5.3] Les *mesures de sécurité* doivent être traitées et surveillées au sein de la *SCL*.

Tableau 2 – Mesures de sécurité déployées pour détecter les erreurs de communication

Erreur de communication	Mesures de sécurité			
	MonitoringNumber[a]	Temporisation avec réception[b]	Ensemble d'ID pour le SafetyProvider[c]	Contrôle d'intégrité des données[d]
Corruption	–	–	–	X
Répétition non prévue	X	X	–	–
Séquence incorrecte	X	–	–	–
Perte	X	X	–	–
Retard inacceptable	–	X	–	–
Insertion	X	–	–	–
Déguisement	X	–	X	X
Adressage	–	–	X	–

[a] Instance de "numéro de séquence" de l'IEC 61784-3.

[b] Instance de "délai" (temporisation) et de "message de réaction" (réception) de l'IEC 61784-3.

[c] Instance d'"authentification de connexion" de l'IEC 61784-3.

[d] Instance d'"assurance d'intégrité des données" de l'IEC 61784-3, fondée sur la signature *CRC*.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Le *SafetyConsumer* est spécifié de sorte que, en cas d'erreur de communication conforme au Tableau 2, une réaction aux anomalies définie se produise.

Dans tous les cas, la *SPDU* défaillante est rejetée et n'est pas transférée à l'application de sécurité.

De plus, si le taux d'erreurs est trop élevé, le *SafetyConsumer* est défini de manière à cesser de fournir des *valeurs de processus* réelles à l'application de sécurité, pour fournir plutôt des *valeurs de substitution Failsafe*. Une indication est également définie au niveau de l'*interface du programme d'application de sécurité*, laquelle peut être interrogée par l'application de sécurité.

Si le taux d'erreurs est toujours jugé acceptable, le diagramme d'états répète la demande (voir 9.4).

5.4 Structure de la couche de communication de sécurité

Le présent document repose sur:

- le *système de transmission normalisé* conforme à l'IEC 62541;
- un protocole de transmission de sécurité supplémentaire qui vient à l'appui de ce *système de transmission normalisé*.

Les applications de sécurité et les applications normalisées partagent les mêmes systèmes de communication IEC 62541 normalisés. La fonction de transmission en toute sécurité comporte des *mesures de sécurité* qui permettent de détecter les anomalies ou les dangers provenant du *système de transmission normalisé*, qui sont susceptibles de compromettre les sous-systèmes de sécurité. Cela comprend les anomalies telles que:

- les erreurs aléatoires (en raison, par exemple, de l'impact du brouillage électromagnétique sur le canal de transmission);
- les défaillances ou anomalies du matériel normalisé;
- les dysfonctionnements systématiques des composants dans le matériel et le logiciel normalisés.

Ce principe permet de limiter l'effort d'évaluation aux "fonctions de transmission en toute sécurité". Il n'est pas exigé de procéder à une évaluation de sécurité fonctionnelle supplémentaire du *système de transmission normalisé*.

Les couches de communication de base du présent document sont représentées à la Figure 2.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

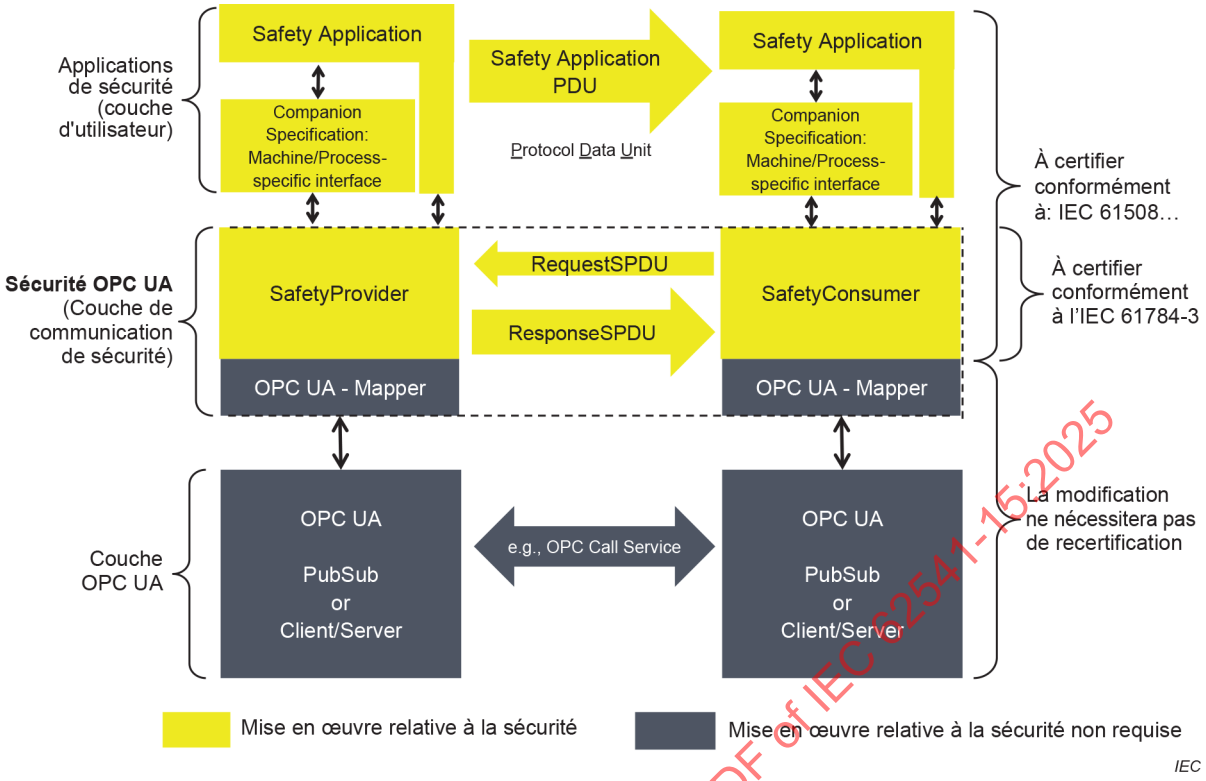


Figure 2 – Architecture de la couche de sécurité

Résumé de l'architecture:

Partie: couche utilisateur

Les applications de sécurité de la couche utilisateur sont directement connectées au *SafetyProvider* ou au *SafetyConsumer*, ou sont connectées par l'intermédiaire d'une interface spécifique à la machine ou au processus, qui est décrite dans les spécifications associées (par exemple, sectorielles).

Les applications de sécurité sont présumées être conçues et mises en œuvre conformément à la série IEC 61508.

Les applications de sécurité de la couche utilisateur ne relèvent pas du domaine d'application du présent document.

Partie: sécurité OPC UA (couche de communication de sécurité)

Cette couche est hors du domaine d'application du présent document. Elle définit les deux services *SafetyProvider* et *SafetyConsumer* en tant que blocs de construction de base. Ensemble, ils forment la *couche de communication de sécurité* (*SCL*), selon une approche de mise en œuvre relative à la sécurité conforme à la série IEC 61508.

Les *SafetyData* sont transmises au moyen d'une communication point à point (unidirectionnelle). Chaque flux de données unidirectionnel communique en interne dans les deux directions, en utilisant un modèle de type demande et réponse. Cela permet de vérifier le caractère opportun des messages en utilisant une seule horloge dans le *SafetyConsumer*, ce qui évite de devoir recourir à des horloges synchronisées.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Lorsque des *SafetyConsumers* se connectent à des *SafetyProviders*, ils ont au préalable des attentes concernant la paire *SafetyProviderID* et *SafetyBaseID* (par exemple, par la configuration). Si le *SafetyProvider* ne comble pas ces attentes, les *valeurs de substitution Failsafe* sont transmises à l'application de sécurité à la place des *valeurs de processus* reçues. En revanche, il n'est pas nécessaire qu'un *SafetyProvider* connaisse le *SafetyConsumerID* du *SafetyConsumer*; il fournit ses *valeurs de processus* à tout *SafetyConsumer* qui en fait la demande.

Les *SafetyProviders* ne sont pas en mesure de détecter les erreurs de communication. Toutes les détections d'erreurs exigées sont réalisées par le *SafetyConsumer*.

S'il est nécessaire qu'une paire d'applications de sécurité échange des SafetyData dans les deux directions, deux paires *SafetyProvider* et *SafetyConsumer* doivent être établies, à raison d'une paire par direction.

Voir l'Article B.2 de l'Annexe B pour des cas d'utilisation décrivant la manière dont *SafetyProviders* et *SafetyConsumers* peuvent être connectés.

Le mappeur OPC UA met en œuvre les parties de la *couche de communication de sécurité* qui sont spécifiques au *Service* de communication de l'IEC 62541 utilisé, c'est-à-dire *PubSub* ou *Client/Serveur*. Par conséquent, les autres parties de la *couche de communication de sécurité* peuvent être mises en œuvre indépendamment du *Service* IEC 62541 utilisé.

Partie: couche OPC UA

Client/Serveur:

- Le SafetyProvider est mis en œuvre à l'aide d'un Serveur IEC 62541 qui fournit une *Méthode*.
- Le *SafetyConsumer* est mis en œuvre à l'aide d'un *Client* IEC 62541 qui appelle la *Méthode* fournie par le *SafetyProvider*.

PubSub:

- Le *SafetyProvider* publie la *ResponseSPDU* et s'abonne à la *RequestSPDU*.
- Le *SafetyConsumer* publie la *RequestSPDU* et s'abonne à la *ResponseSPDU*.

5.5 Exigences relatives au calcul du CRC

[RQ5.4] Tout calcul de la signature *CRC* doit commencer par une valeur prédéfinie de "1".

[RQ5.5] Tout calcul de la signature *CRC* qui a pour résultat une valeur de "0" doit utiliser la valeur "1" à la place.

[RQ5.6] Les *SPDU* dont toutes les valeurs (y compris la signature *CRC*) sont nulles doivent être ignorées par le récepteur (*SafetyConsumer* et *SafetyProvider*).

6 Services de la couche de communication de sécurité

6.1 Généralités

L'Article 6 décrit l'intégration du présent document dans les *Modèles d'information* IEC 62541 au 6.2, et les interfaces de *Service* qui en résultent au 6.3. Les services de diagnostic sont décrits au 6.4.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

6.2 Modèles d'information

6.2.1 Généralités

Le 6.2 décrit les identificateurs, les types et la structure des *Objets* et *Méthodes* utilisés pour mettre en œuvre les *mappeurs OPC UA* définis dans le présent document. Cette mise en œuvre a trois objectifs:

- la prise en charge de l'échange sécurisé de *SPDU* pendant l'exécution;
- la navigation en ligne, pour identifier les *SafetyConsumers* et les *SafetyProviders*, et pour vérifier leurs paramètres à des fins de diagnostic;
- l'ingénierie hors ligne: le *Modèle d'information* d'un contrôleur peut être exporté dans un fichier normalisé sur son système d'ingénierie, puis être importé dans un autre système d'ingénierie, avant d'être finalement déployé sur un autre contrôleur. Cela permet d'échanger les interfaces de communication des applications de sécurité indépendamment du fournisseur, par exemple pour établir des connexions entre des appareils.

NOTE Ni la navigation en ligne ni l'ingénierie hors ligne ne prennent actuellement en charge des fonctionnalités de détection d'erreurs. Par conséquent, il n'y a aucune garantie de sécurité fonctionnelle. Cela signifie que la navigation en ligne ne peut être utilisée qu'à des fins de diagnostic et non pour échanger des données relatives à la sécurité. Il est admis par hypothèse que des erreurs peuvent se produire lors du transfert du *Modèle d'information* d'un système d'ingénierie à un autre dans le contexte de l'ingénierie hors ligne. Le programmeur de l'application de sécurité est donc responsable de la vérification et de la validation de l'application de sécurité.

Par conséquent, toutes les valeurs de type décrites au 6.2 sont définies en lecture seule, c'est-à-dire qu'elles ne peuvent pas être écrites par des commandes d'écriture générales de l'IEC 62541.

6.2.2 Définition des objets et des ObjectTypes

6.2.2.1 Objet SafetyACSet

[RQ6.1] Chaque *Serveur* doit avoir un *Dossier* singleton appelé *SafetyACSet*, avec un *NodeID* fixe dans l'*Espace de noms* du présent document. Étant donné que tous les *SafetyProviders* et *SafetyConsumers* de ce *Serveur* contiennent une *Référence* hiérarchique entre cet *Objet* et eux-mêmes, ils peuvent être utilisés pour accéder directement à tous les *SafetyProviders* et/ou *SafetyConsumers*. *SafetyACSet* est destiné à des fins de sécurité uniquement. Il convient qu'il ne fasse pas référence à des éléments *non relatifs à la sécurité*.

Voir le Tableau 3 pour la définition du *SafetyACSet*.

Tableau 3 – Définition de SafetyACSet

Attribut	Valeur		
BrowseName	SafetyACSet		
References	**NodeClass**	**BrowseName**	**Commentaire**
OrganizedBy par le Dossier Objects défini dans l'IEC 62541-5.			
HasTypeDefinition	ObjectType	FolderType	Point d'entrée pour tous les *SafetyProviders* et *SafetyConsumers*
Unités de conformité			
SafetyACSet			

[RQ6.2] En outre, un *Serveur* doit comprendre un *Objet* IEC 62541 issu du *DataType SafetyProviderType* pour chaque *SafetyProvider* qu'il met en œuvre, ainsi qu'un *Objet* IEC 62541 issu du *DataType SafetyConsumerType* pour chaque *SafetyConsumer* qu'il met en œuvre. Les *Modèles d'information* correspondants représentés à la Figure 3 et à la Figure 4 doivent être utilisés.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Une description de la notation graphique des différents types de nœuds et de références (voir Figure 3, Figure 4 et Figure 6) peut être consultée dans l'IEC 62541-3.

La Figure 3 décrit le *SafetyProvider* et le *SafetyConsumer*.

NOTE 1 Le présent document prend pour hypothèse un échange de données cohérent (atomique) entre les mappeurs OPC des deux points d'extrémité.

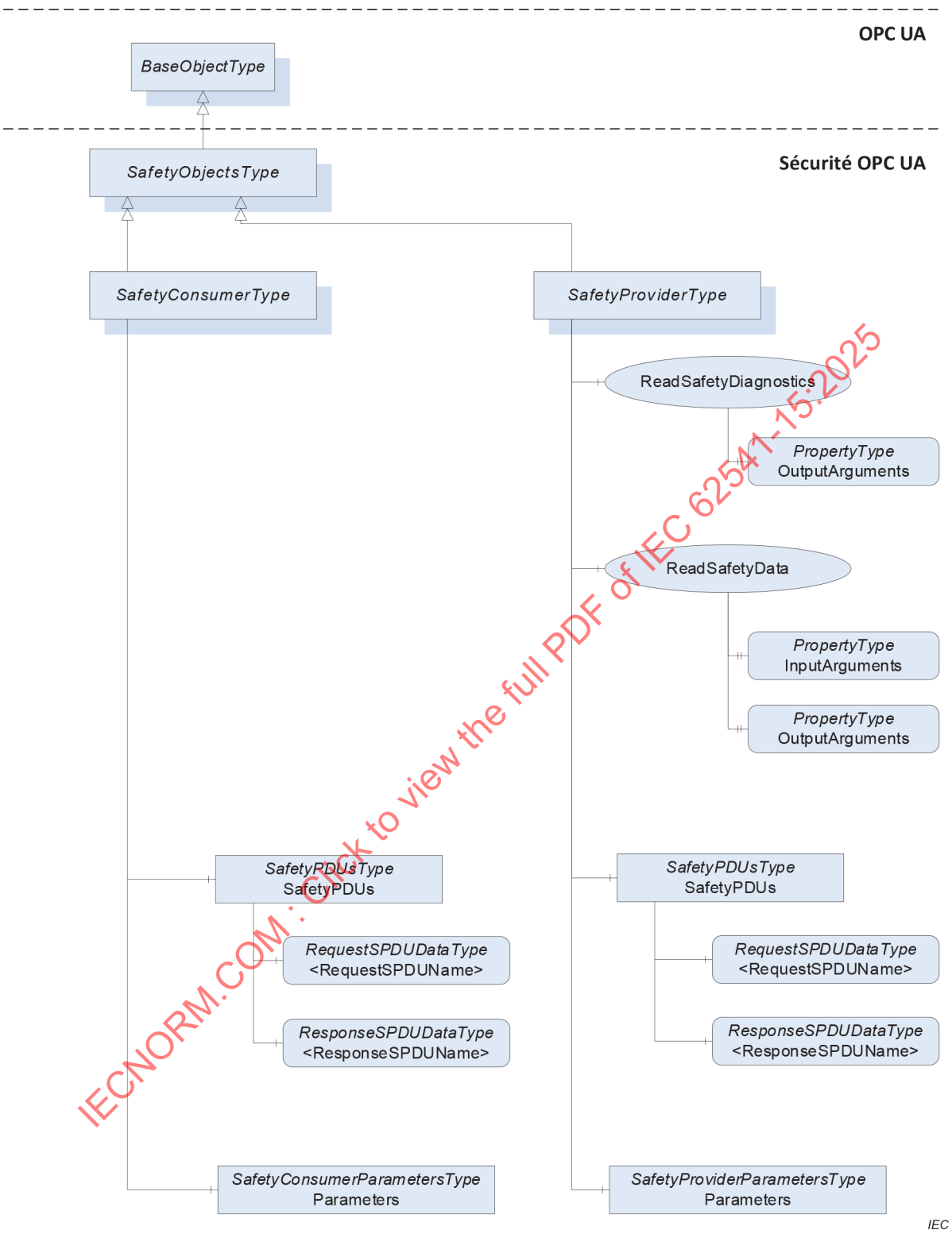
[RQ6.3a] Pour les mises en œuvre qui prennent en charge le *Client/Serveur* IEC 62541, le *Service Call* du *Jeu de services Method* (voir l'IEC 62541-4) doit être utilisé. La *Méthode ReadSafetyData* comporte un ensemble d'arguments d'entrée qui forment la *RequestSPDU* et un ensemble d'arguments de sortie qui forment la *ResponseSPDU*. Le *SafetyConsumer* utilise le Client IEC 62541 avec l'*Appel de Service* IEC 62541.

[RQ6.3b] Pour les mises en œuvre qui prennent en charge le *PubSub* IEC 62541, l'*Objet SafetyPDUs* de l'IEC 62541 avec ses *Propriétés RequestSPDU* et *ResponseSPDU* doit être utilisé. La *RequestSPDU* est publiée par le *SafetyConsumer* et souscrite par le *SafetyProvider*. La *ResponseSPDU* est publiée par le *SafetyProvider* et souscrite par le *SafetyConsumer*.

NOTE 2 Les termes "request" (demande) et "response" (réponse) font référence au comportement sur la couche du présent document. Dans le contexte du *PubSub*, les demandes et les réponses sont réalisées par la publication et la souscription répétées de *Messages* (voir Figure 14).

[RQ6.4] À des fins de diagnostic, les *SPDU* reçues et envoyées doivent être accessibles en appelant la *Méthode ReadSafetyDiagnostics*.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

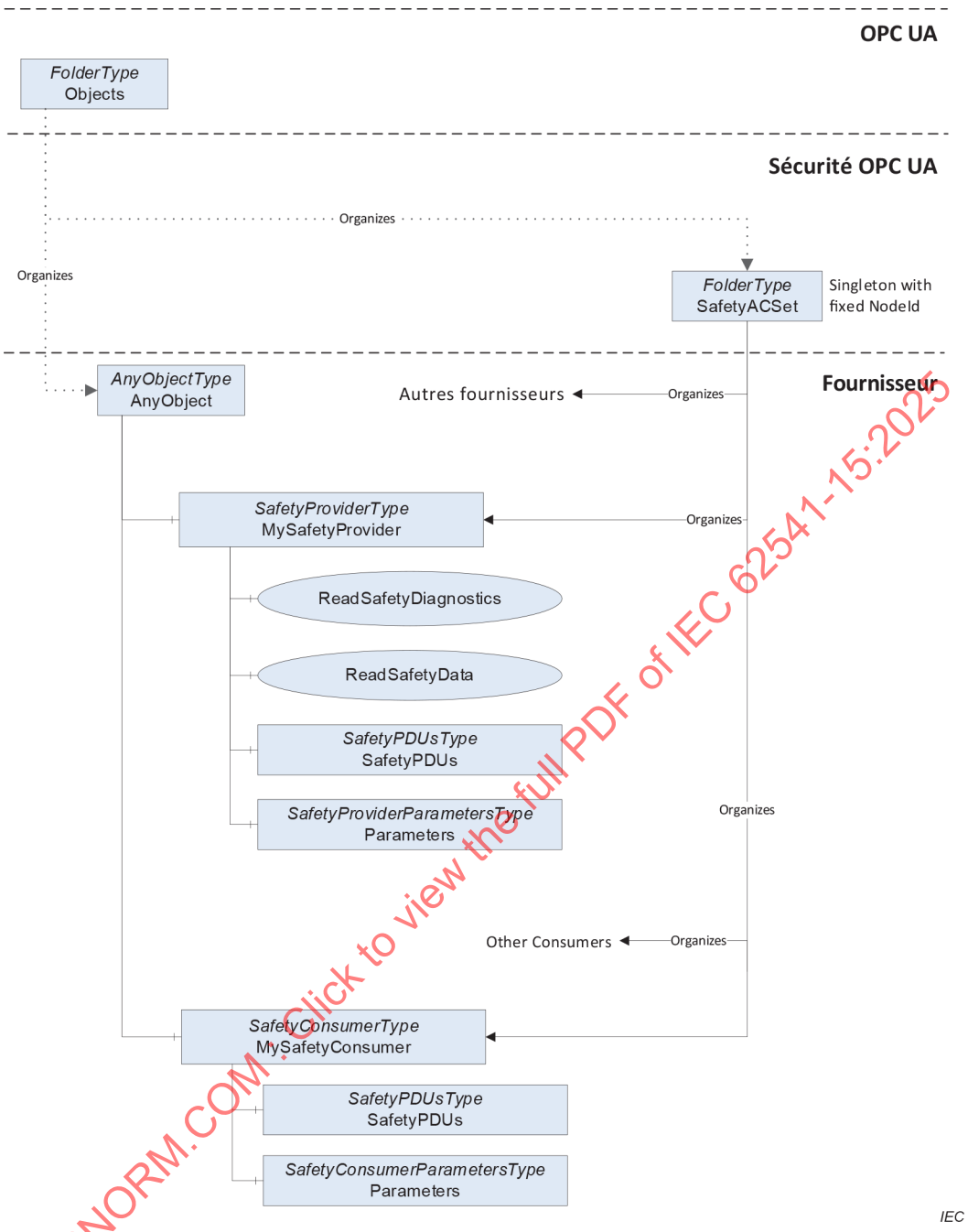


IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Figure 3 – Objets Serveur pour la sécurité OPC UA

NOTE 3 Pour les arguments d'entrée/sortie (I/O, *Input/Output*) des *Méthodes ReadSafetyData* et *ReadSafetyDiagnostics*, voir le 6.2.2.3 et le 6.2.2.4. Pour les paramètres du *SafetyProvider* et du *SafetyConsumer*, voir la Figure 6, le Tableau 12 et le Tableau 13. Pour la *RequestSPDU* et la *ResponseSPDU*, voir le Tableau 7, le Tableau 18, le Tableau 20 et le 7.2.1.

La Figure 4 représente les instances d'*Objets Serveur* pour le présent document. L'*ObjectType* du *SafetyProviderType* contient des *Méthodes* comportant des sorties de la *Structure* du *DataType* abstrait. Chaque instance d'un *SafetyProvider* exige sa propre copie des *Méthodes* qui contiennent les *DataTypes* concrets pour *OutSafetyData* et *OutNonSafetyData*.



IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Figure 4 – Instances d'objets de serveur pour le présent document

6.2.2.2 Définition de l'ObjectType Safety

[RQ6.5] Pour réduire le nombre de variations et alléger les essais de validation, les restrictions suivantes s'appliquent aux instances de *SafetyProviderType* et de *SafetyConsumerType* (ou aux instances de *DataTypes* issus de *SafetyProviderType* ou de *SafetyConsumerType*):

1) les références représentées à la Figure 4 provenant de *SafetyProviderType* ou *SafetyConsumerType* et des niveaux inférieurs doivent être de *ReferenceType HasComponent* (et ne doivent pas être issues du *ReferenceType HasComponent*) pour les *Références d'objets* ou de *ReferenceType HasProperty* (et ne doivent pas être issues du *ReferenceType HasProperty*) pour les *Références de propriétés*;
2) étant donné que les *BrowseNames* (c'est-à-dire le nom et l'*Espace de noms*) sont utilisés pour trouver des *Méthodes*, les noms d'*Objets* et de *Propriétés* doivent être uniques au niveau local;

3) le *DataType* des *Propriétés* et des *MethodArguments* doit être utilisé comme cela spécifié, et aucun *DataType* dérivé ne doit être utilisé (exception: *OutSafetyData* et *OutNonSafetyData*);

4) dans l'IEC 62541, l'ordre des arguments de *Méthode* est pertinent.

Voir le Tableau 4 pour la définition du *SafetyObjectsType*.

Tableau 4 – Définition de SafetyObjectsType

Attribut	Valeur				
BrowseName	SafetyObjectsType				
IsAbstract	True				
References	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Sous-type de BaseObjectType					
Unités de conformité					
SafetySupport					

Voir le Tableau 5 pour la définition du *SafetyProviderType*.

Tableau 5 – Définition de SafetyProviderType

Attribut	Valeur				
BrowseName	SafetyProviderType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Sous-type de SafetyObjectsType					
HasComponent	Method	ReadSafetyData			Optional
HasComponent	Method	ReadSafetyDiagnostics			Optional
HasComponent	Object	SafetyPDUs		SafetyPDUsType	Optional
HasComponent	Object	Parameters		SafetyProviderParameters Type	Mandatory
Unités de conformité					
SafetyProviderParameters					

[RQ6.6] Les instances de *SafetyProviderType* doivent utiliser des *DataTypes* non abstraits pour les arguments *OutSafetyData* et *OutNonSafetyData*.

Voir le Tableau 6 pour la définition du *SafetyConsumerType*.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Tableau 6 – Définition de SafetyConsumerType

Attribut	Valeur				
BrowseName	SafetyConsumerType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling Rule**
Sous-type de SafetyObjectsType					
HasComponent	Object	SafetyPDUs		SafetyPDUsType	Optional
HasComponent	Object	Parameters		SafetyConsumer-ParametersType	Mandatory
Unités de conformité					
SafetyConsumerParameters					

6.2.2.3 Méthode ReadSafetyData

Cette *Méthode* est obligatoire pour la *Facette SafetyProviderServerMapper*. Elle est utilisée pour lire les *SafetyData* du *SafetyProvider*. Il incombe à l'application de sécurité de s'assurer que cette *Méthode* n'est pas simultanément appelée par plusieurs *SafetyConsumers*. Dans le cas contraire, le *SafetyConsumer* peut recevoir des réponses non valides donnant lieu à une réaction de sécurité qui peut conduire à des déclenchements parasites et/ou à une indisponibilité du système.

Voir le Tableau 7 pour les arguments de la *Méthode ReadSafetyData* et le Tableau 8 pour la définition de son *AddressSpace*.

L'argument de *Méthode OutSafetyData* possède un *DataType* spécifique à l'application issu de *Structure*. Ce *DataType* (y compris le *DataTypeID*) est présumé identique dans le *SafetyProvider* et dans le *SafetyConsumer*. Sinon, le *SafetyConsumer* n'accepte pas les données transférées et passe à des *valeurs de substitution Failsafe* (voir l'état S16 du Tableau 34 ainsi que le 7.2.3.2 et le 7.2.3.5). L'argument de *Méthode OutNonSafetyData* possède un *DataType* spécifique à l'application issu de *Structure*.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Signature

```
ReadSafetyData (
    [in]   UInt32                InSafetyConsumerID,
    [in]   UInt32                InMonitoringNumber,
    [in]   InFlagsType           InFlags,
    [out]  Structure             OutSafetyData,
    [out]  OutFlagsType          OutFlags,
    [out]  UInt32                OutSPDU_ID_1,
    [out]  UInt32                OutSPDU_ID_2,
    [out]  UInt32                OutSPDU_ID_3,
    [out]  UInt32                OutSafetyConsumerID,
    [out]  UInt32                OutMonitoringNumber,
    [out]  UInt32                OutCRC,
    [out]  Structure             OutNonSafetyData)
;
```

Tableau 7 – Arguments de la méthode ReadSafetyData

Argument	Description
InSafetyConsumerID	"Identificateur du consommateur de sécurité" (voir *SafetyConsumerID* dans le Tableau 23).
InMonitoringNumber	"*MonitoringNumber* de la *RequestSPDU*" (voir 7.2.1.3 et *MonitoringNumber* dans le Tableau 23).
InFlags	"Octet avec *fanions non relatifs à la sécurité* issus de *SafetyConsumer*" (voir 6.2.3.1).
OutSafetyData	"*SafetyData*" (voir 7.2.1.5).
OutFlags	"Octet avec *fanions* relatifs à la sécurité issus de *SafetyProvider*" (voir 6.2.3.2).
OutSPDU_ID_1	"Identificateur de la PDU de sécurité Partie 1" (voir 7.2.3.2).
OutSPDU_ID_2	"Identificateur de la PDU de sécurité Partie 2" (voir 7.2.3.2).
OutSPDU_ID_3	"Identificateur de la PDU de sécurité Partie 3" (voir 7.2.3.2).
OutSafetyConsumerID	"Identificateur du consommateur de sécurité" (voir *SafetyConsumerID* dans le Tableau 23 et le Tableau 26).
OutMonitoringNumber	*MonitoringNumber* de la *ResponseSPDU* (voir 7.2.1.9, 7.2.3.1 et Figure 11).
OutCRC	*CRC* sur la *ResponseSPDU* (voir 7.2.3.6).
OutNonSafetyData	"Données non sécurisées" (voir 7.2.1.11).

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Tableau 8 – Définition de l'AddressSpace pour la méthode ReadSafetyData

Attribut	Valeur				
BrowseName	ReadSafetyData				
References	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
HasProperty	Variable	InputArguments	Argument[]	PropertyType	Mandatory
HasProperty	Variable	OutputArguments	Argument[]	PropertyType	Mandatory
Unités de conformité					
ReadSafetyData					

6.2.2.4 Méthode ReadSafetyDiagnostics

Cette *Méthode* est obligatoire pour la *Facette SafetyProviderServerMapper* et facultative pour la *Facette SafetyProviderPubSubMapper*. Elle est fournie pour chaque *SafetyProvider* qui sert d'*interface de diagnostic* (voir 6.4.3).

Voir le Tableau 9 pour les arguments de la *Méthode ReadSafetyDiagnostics* et le Tableau 10 pour la définition de son *AddressSpace*.

Les arguments de la *Méthode OutSafetyData* et *OutNonSafetyData* possèdent des types spécifiques à l'application issus de *Structure*.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Signature

```
ReadSafetyDiagnostics (
    [out]  UInt32                  InSafetyConsumerID,
    [out]  UInt32                  InMonitoringNumber,
    [out]  InFlagsType             InFlags,
    [out]  Structure               OutSafetyData,
    [out]  OutFlagsType            OutFlags,
    [out]  UInt32                  OutSPDU_ID_1,
    [out]  UInt32                  OutSPDU_ID_2,
    [out]  UInt32                  OutSPDU_ID_3,
    [out]  UInt32                  OutSafetyConsumerID,
    [out]  UInt32                  OutMonitoringNumber,
    [out]  UInt32                  OutCRC,
    [out]  Structure               OutNonSafetyData)
;
```

Tableau 9 – Arguments de la méthode ReadSafetyDiagnostics

Argument	Description
InSafetyConsumerID	Voir le Tableau 7
InMonitoringNumber	Voir le Tableau 7
InFlags	Voir le Tableau 7
OutSafetyData	Voir le Tableau 7
OutFlags	Voir le Tableau 7
OutSPDU_ID_1	Voir le Tableau 7
OutSPDU_ID_2	Voir le Tableau 7
OutSPDU_ID_3	Voir le Tableau 7
OutSafetyConsumerID	Voir le Tableau 7
OutMonitoringNumber	Voir le Tableau 7
OutCRC	Voir le Tableau 7
OutNonSafetyData	Voir le Tableau 7

Tableau 10 – Définition de l'AddressSpace pour la méthode ReadSafetyDiagnostics

Attribut	Valeur				
BrowseName	ReadSafetyDiagnostics				
References	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
HasProperty	Variable	OutputArguments	Argument[]	PropertyType	Mandatory
Unités de conformité					
ReadSafetyDiagnostics					

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

6.2.2.5 Objet SafetyPDUs

Cet *Objet* est obligatoire pour la *Facette SafetyProviderPubSubMapper* et la *Facette SafetyConsumerPubSubMapper*. Il est utilisé par le *SafetyProvider* pour s'abonner à la *RequestSPDU* et publier la *ResponseSPDU*. Le *DataType* de la *RequestSPDU* est structuré de la même manière que les arguments d'entrée de *ReadSafetyData*. Le *DataType* de la *ResponseSPDU* est structuré de la même manière que les arguments de sortie de *ReadSafetyData*.

Voir le Tableau 11 pour la définition du *ConfigurableObjectType*.

Les deux variables du *SafetyPDUsType* ont un homologue dans le *Modèle d'information* du *SafetyConsumer*. Le *SafetyConsumer* publie la *RequestSPDU* et s'abonne à la *ResponseSPDU*.

Tableau 11 – Définition de SafetyPDUsType

Attribut	Valeur				
BrowseName	SafetyPDUsType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling Rule**
Sous-type de BaseObjectType					
HasComponent	Variable	<RequestSPDU>	RequestSPDUDataType	BaseDataVariableType	Espace réservé - obligatoire
HasComponent	Variable	<ResponseSPDU>	ResponseSPDUDataType	BaseDataVariableType	Espace réservé - obligatoire
Unités de conformité					
SafetyPDUs					

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

L'*Objet SafetyPDUs* doit contenir exactement une *Référence* à une *Variable* d'un *DataType RequestSPDUDataType* et exactement une *Référence* à une *Variable* d'un sous-type du *DataType ResponseSPDUDataType*.

Par exemple, la Figure 5 représente une application de sécurité distribuée avec quatre *SafetyAutomationComponents*. Il est admis par hypothèse que le *SafetyAutomationComponent 1* envoie une valeur aux trois autres *SafetyAutomationComponents* en utilisant trois *SafetyProviders*, chacun comprenant une paire de *SPDU*. À chaque destinataire correspond une paire individuelle de *SPDU*.

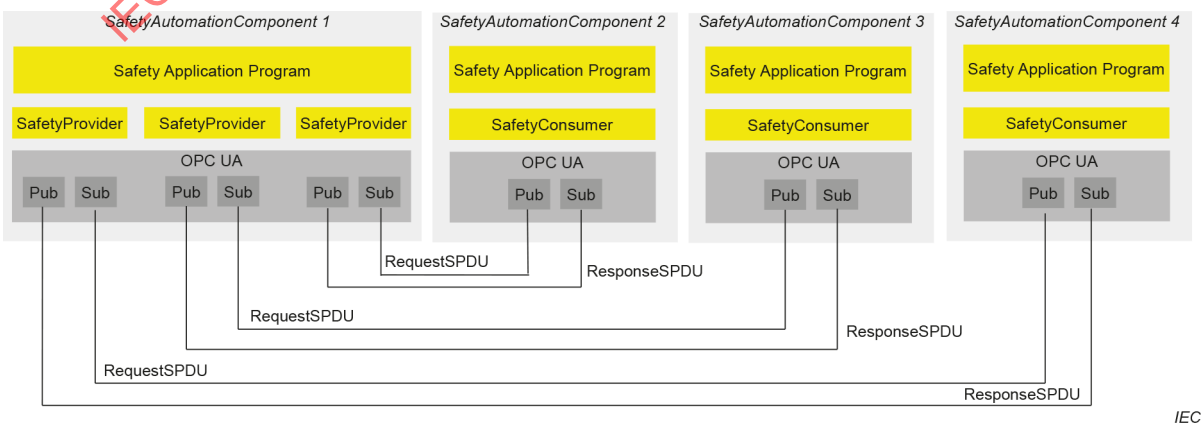
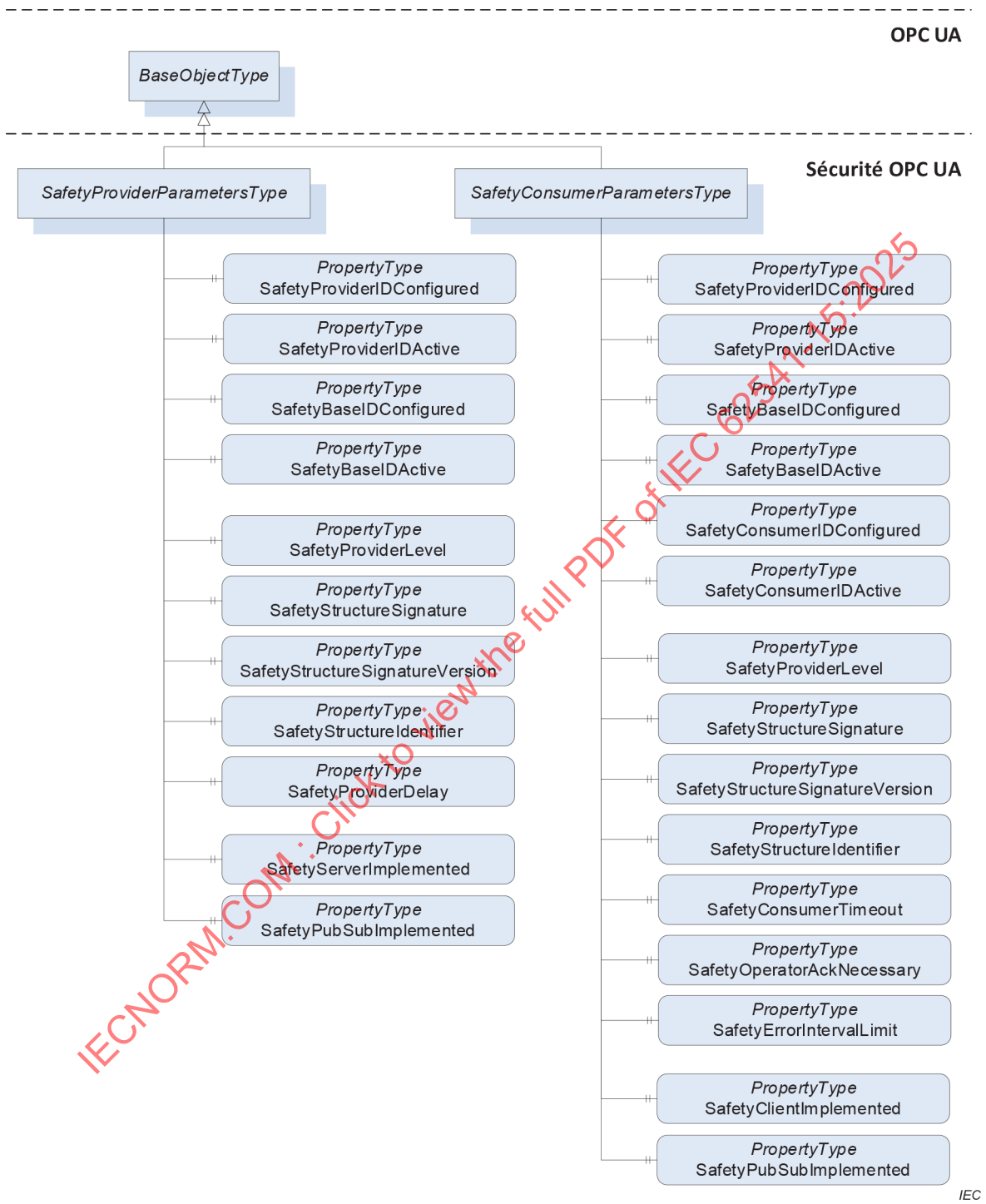


Figure 5 – Multidiffusion de sécurité avec trois destinataires utilisant le PubSub IEC 62541

6.2.2.6 Objets SafetyProviderParameters et SafetyConsumerParameters

La Figure 6 représente les paramètres de sécurité pour le *SafetyProvider* et le *SafetyConsumer*.



IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Figure 6 – Paramètres de sécurité du SafetyProvider et du SafetyConsumer

Le Tableau 12 donne la définition du *SafetyProviderParametersType*. Voir le 6.3.3.3 pour plus de détails sur l'*interface de paramètres de sécurité* (*SPI*) du *SafetyProvider*.

Tableau 12 – Définition de SafetyProviderParametersType

Attribut	Valeur				
BrowseName	SafetyProviderParametersType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling Rule**
Sous-type de BaseObjectType					
HasProperty	Variable	SafetyProviderIDConfigured	UInt32	PropertyType	Mandatory
HasProperty	Variable	SafetyProviderIDActive	UInt32	PropertyType	Mandatory
HasProperty	Variable	SafetyBaseIDConfigured	Guid	PropertyType	Mandatory
HasProperty	Variable	SafetyBaseIDActive	Guid	PropertyType	Mandatory
HasProperty	Variable	SafetyProviderLevel	Byte	PropertyType	Mandatory
HasProperty	Variable	SafetyStructureSignature	UInt32	PropertyType	Mandatory
HasProperty	Variable	SafetyStructureSignatureVersion	UInt16	PropertyType	Mandatory
HasProperty	Variable	SafetyStructureIdentifier	String	PropertyType	Mandatory
HasProperty	Variable	SafetyProviderDelay	UInt32	PropertyType	Mandatory
HasProperty	Variable	SafetyServerImplemented	Boolean	PropertyType	Mandatory
HasProperty	Variable	SafetyPubSubImplemented	Boolean	PropertyType	Mandatory
Unités de conformité					
SafetyProviderParameters					

Les paramètres de *SafetyProviderID* et de *SafetyBaseID* sont disponibles sous forme de paires pour les états "Configured" et "Active":

- *SafetyProviderIDConfigured* et *SafetyProviderIDActive*;
- *SafetyBaseIDConfigured* et *SafetyBaseIDActive*.

Les paramètres "[...]Configured" doivent toujours fournir les valeurs configurées par l'intermédiaire de la *SPI*. Les paramètres "[...]Active" doivent fournir:

- les valeurs "[...]Configured" correspondantes si le système est toujours hors ligne;
- les valeurs qui ont été définies pendant l'exécution par les paramètres de la *SAPI* (*SafetyProviderID*, *SafetyBaseID*);
- les valeurs "[...]Configured" correspondantes si les valeurs actives ont été mises à zéro par les paramètres de la *SAPI* (*SafetyProviderID*, *SafetyBaseID*).

La *Propriété SafetyBaseIDConfigured* est partagée pour tous les *SafetyProviders* ayant la même valeur de *SafetyBaseIDConfigured*. Si plusieurs instances de *SafetyObjectsType* sont en cours d'exécution sur le même *Nœud*, il est bénéfique qu'une *Propriété SafetyBaseIDConfigured* soit référencée par plusieurs *SafetyProviders* et/ou *SafetyConsumers*.

Jusqu'à la version 2.0 du document, la valeur de la *SafetyStructureSignatureVersion* doit être 0x0001 (voir RQ7.21 au 7.2.3.5).

Le Tableau 13 fournit une définition du *SafetyConsumerParametersType*. Les *Propriétés SafetyStructureIdentifier* et *SafetyStructureSignatureVersion* sont facultatives, car la *SafetyStructureSignature* est généralement calculée dans un outil d'ingénierie hors ligne. Pour les petits appareils, il peut être avantageux de télécharger uniquement la *SafetyStructureSignature* sur l'appareil, mais pas le *SafetyStructureIdentifier* ni la *SafetyStructureSignatureVersion*, afin d'économiser de la bande passante et/ou de la mémoire. Voir le 6.3.4.4 pour plus de détails sur l'*interface de paramètres de sécurité* (*SPI*) du *SafetyConsumer*.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Tableau 13 – Définition de SafetyConsumerParametersType

Attribut	Valeur				
BrowseName	SafetyConsumerParametersType				
IsAbstract	False				
References	**Node Class**	**BrowseName**	**DataType**	**TypeDefinition**	**Modelling Rule**
Sous-type de BaseObjectType					
HasProperty	Variable	SafetyProviderIDConfigured	UInt32	PropertyType	Mandatory
HasProperty	Variable	SafetyProviderIDActive	UInt32	PropertyType	Mandatory
HasProperty	Variable	SafetyBaseIDConfigured	Guid	PropertyType	Mandatory
HasProperty	Variable	SafetyBaseIDActive	Guid	PropertyType	Mandatory
HasProperty	Variable	SafetyConsumerIDConfigured	UInt32	PropertyType	Mandatory
HasProperty	Variable	SafetyConsumerIDActive	UInt32	PropertyType	Mandatory
HasProperty	Variable	SafetyProviderLevel	Byte	PropertyType	Mandatory
HasProperty	Variable	SafetyStructureSignature	UInt32	PropertyType	Mandatory
HasProperty	Variable	SafetyStructureSignatureVersion	UInt16	PropertyType	Optional
HasProperty	Variable	SafetyStructureIdentifier	String	PropertyType	Optional
HasProperty	Variable	SafetyConsumerTimeout	UInt32	PropertyType	Mandatory
HasProperty	Variable	SafetyOperatorAckNecessary	Boolean	PropertyType	Mandatory
HasProperty	Variable	SafetyErrorIntervalLimit	UInt16	PropertyType	Mandatory
HasProperty	Variable	SafetyClientImplemented	Boolean	PropertyType	Mandatory
HasProperty	Variable	SafetyPubSubImplemented	Boolean	PropertyType	Mandatory
Unités de conformité					
SafetyConsumerParameters					

Les paramètres de *SafetyProviderID*, *SafetyBaseID* et *SafetyConsumerID* sont disponibles sous forme de paires pour les états "Configured" et "Active": *SafetyProviderIDConfigured* et *SafetyProviderIDActive*, *SafetyBaseIDConfigured* et *SafetyBaseIDActive*, et *SafetyConsumerIDConfigured* et *SafetyConsumerIDActive*.

Les paramètres "[...]Configured" doivent toujours fournir les valeurs configurées par l'intermédiaire de la *SPI*. Les paramètres "[...]Active" doivent fournir:

- les valeurs "[...]Configured" correspondantes si le système est toujours hors ligne;
- les valeurs qui ont été définies pendant l'exécution par les paramètres de la *SAPI* (*SafetyProviderID*, *SafetyBaseID*, *SafetyConsumerID*);
- les valeurs "[...]Configured" correspondantes si les valeurs actives ont été mises à zéro par les paramètres de la *SAPI* (*SafetyProviderID*, *SafetyBaseID*, *SafetyConsumerID*).

6.2.3 Définition de DataType

6.2.3.1 InFlagsType

L'InFlagsType est un sous-type du *DataType Byte* avec la *Propriété OptionSetValues* définie. L'*InFlagsType* est défini de manière formelle dans le Tableau 14.

CommunicationError peut être utilisé comme déclencheur, par exemple pour un outil d'analyse de communication. Il n'est pas transféré à l'application de sécurité par le *SafetyProvider*. Si CommunicationError est nécessaire dans l'application de sécurité, une communication bidirectionnelle peut être mise en œuvre et la valeur de CommunicationError peut être insérée dans les données utilisateur.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Tableau 14 – Valeurs d'InFlagsType

Valeur	N° de bit	Description
CommunicationError	0	0: pas d'erreur 1: une erreur a été détectée dans la *ResponseSPDU* précédente.
OperatorAckRequested	1	Utilisé pour informer le *SafetyProvider* qu'un acquittement de l'opérateur est demandé.
FSV_Activated	2	Utilisé pour l'essai de conformité de SafetyConsumer.SAPI.FSV_Activated

Les bits 3 à 7 sont réservés pour une utilisation future et doivent être mis à zéro par le *SafetyConsumer*. Ils ne doivent pas être évalués par le *SafetyProvider*.

La représentation d'*InFlagsType* dans l'*AddressSpace* est définie dans le Tableau 15.

Tableau 15 – Définition d'InFlagsType

Attribut	Valeur				
BrowseName	InFlagsType				
IsAbstract	False				
References	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**Autre**
Sous-type du *DataType Byte* défini dans l'IEC 62541-3.					
0:HasProperty	Variable	0:OptionSetValues	0:LocalizedText []	0:PropertyType	
Unités de conformité					
SafetySupport					

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

6.2.3.2 OutFlagsType

L'*OutFlagsType* est un sous-type du *DataType Byte* avec la *Propriété OptionSetValues* définie. L'*OutFlagsType* est défini de manière formelle dans le Tableau 16.

Tableau 16 – Valeurs d'OutFlagsType

Valeur	N° de bit	Description
OperatorAckProvider	0	Acquittement de l'opérateur auprès du fournisseur, donc transféré au *SafetyConsumer* (voir OperatorAckProvider dans la *SAPI* du *SafetyProvider*, 6.3.3.2)
ActivateFSV	1	Activation des valeurs *Failsafe* par l'application de sécurité auprès du *SafetyProvider*, donc transférées au *SafetyConsumer* (voir *ActivateFSV* dans la *SAPI* du *SafetyProvider*, 6.3.3.2
TestModeActivated	2	Activation et désactivation du mode d'essai dans le *SafetyProvider*, donc transféré au *SafetyConsumer* (voir *EnableTestMode* dans la *SAPI* du *SafetyProvider*, 6.3.3.2

Les bits 3 à 7 sont réservés pour une utilisation future et doivent être mis à zéro par le *SafetyProvider*. Ils ne doivent pas être évalués par le *SafetyConsumer*.

La représentation d'*OutFlagsType* dans l'*AddressSpace* est définie dans le Tableau 17.

Tableau 17 – Définition d'OutFlagsType

Attribut	Valeur				
BrowseName	OutFlagsType				
IsAbstract	False				
References	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**Autre**
Sous-type du *DataType Byte* défini dans l'IEC 62541-3.					
0:HasProperty	Variable	0:OptionSetValues	0:LocalizedText []	0:PropertyType	
Unités de conformité					
SafetySupport					

6.2.3.3 RequestSPDUDataType

Le Tableau 18 fournit une définition du *RequestSPDUDataType*. Le Préfixe "In" est interprété du point de vue du *SafetyProvider* et est utilisé de manière cohérente avec les paramètres de la *Méthode ReadSafetyData* (voir 6.2.2.3).

Tableau 18 – Structure de RequestSPDUDataType

Nom	Type	Description
RequestSPDUDataType	structure	
InSafetyConsumerID	UInt32	Voir l'argument de *Méthode* correspondant dans le Tableau 7.
InMonitoringNumber	UInt32	Voir l'argument de *Méthode* correspondant dans le Tableau 7.
InFlags	InFlagsType	Voir l'argument de *Méthode* correspondant dans le Tableau 7.

La représentation du *RequestSPDUDataType* dans l'*AddressSpace* est définie dans le Tableau 19.

Tableau 19 – Définition de RequestSPDUDataType

Attributs	Valeur				
BrowseName	**RequestSPDUDataType**				
IsAbstract	False				
References	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Sous-type de *Structure* défini dans l'IEC 62541-3.					
Unités de conformité					
SafetyPDUs					

6.2.3.4 ResponseSPDUDataType

Le Tableau 20 décrit la *Structure* du *ResponseSPDUDataType*. Le Préfixe "Out" est interprété du point de vue du *SafetyProvider* et est utilisé de manière cohérente avec les paramètres de la *Méthode ReadSafetyData* (voir 6.2.2.3).

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Tableau 20 – Structure de ResponseSPDUDataType

Nom	Type	Description
ResponseSPDUDataType	structure	
OutFlags	OutFlagsType	Voir l'argument de *Méthode* correspondant dans le Tableau 7.
OutSPDU_ID_1	UInt32	Voir l'argument de *Méthode* correspondant dans le Tableau 7.
OutSPDU_ID_2	UInt32	Voir l'argument de *Méthode* correspondant dans le Tableau 7.
OutSPDU_ID_3	UInt32	Voir l'argument de *Méthode* correspondant dans le Tableau 7.
OutSafetyConsumerID	UInt32	Voir l'argument de *Méthode* correspondant dans le Tableau 7.
OutMonitoringNumber	UInt32	Voir l'argument de *Méthode* correspondant dans le Tableau 7.
OutCRC	UInt32	Voir l'argument de *Méthode* correspondant dans le Tableau 7.

[RQ6.7] Pour définir le *DataType* concret de la *ResponseSPDU* (qui spécifie les *DataTypes* concrets de *SafetyData* et de *NonSafetyData*, respectivement), procéder comme suit: (1) déterminer un *DataType* concret à partir du *ResponseSPDUDataType* abstrait; (2) ce faisant, ajouter les champs suivants à la *Structure* dans l'ordre indiqué: (a) premièrement, le champ *OutSafetyData* avec le *DataType Structure* concret pour *SafetyData* (voir 7.2.1.5); (b) deuxièmement, le champ *NonSafetyData* avec le *DataType Structure* concret pour *NonSafetyData* (ou un *DataType* d'espace réservé; voir l'exigence RQ6.8).

[RQ6.8] Pour éviter d'éventuels problèmes avec des *Structures* vides, la *Structure* factice *NonSafetyDataPlaceholder* doit être utilisée comme *DataType* pour *OutNonSafetyData* lorsqu'aucune *NonSafetyData* n'est utilisée. Le *DataType Node* qui définit cette *Structure* a un *NodeID* fixe et contient une valeur *booléenne* unique.

La représentation du *ResponseSPDUDataType* dans l'*AddressSpace* est définie dans le Tableau 21.

Tableau 21 – Définition de ResponseSPDUDataType

Attributs	Valeur				
BrowseName	ResponseSPDUDataType				
IsAbstract	True				
References	**NodeClass**	**BrowseName**	**DataType**	**TypeDefinition**	**ModellingRule**
Sous-type de *Structure* défini dans l'IEC 62541-3.					
Unités de conformité					
SafetyPDUs					

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

6.2.3.5 NonSafetyDataPlaceholderDataType

Le Tableau 22 fournit une définition du *NonSafetyDataPlaceholderDataType*. Le récepteur ne doit pas évaluer la valeur de "Dummy".

Tableau 22 – Structure de NonSafetyDataPlaceholderDataType

Nom	Type	Description
NonSafetyDataPlaceholderDataType	Structure	
Dummy	Boolean	*Variable* factice pour éviter les structures vides.

6.2.4 Version de SafetyProvider

Les versions futures peuvent utiliser différents identificateurs (tels que *ReadSafetyDataV2* pour la *Méthode* lorsque la communication *Client/Serveur* est utilisée ou *RequestSPDUV2DataType* et *ResponseSPDUV2DataType* pour les *DataTypes* de *SPDU* lorsque la communication *PubSub* est utilisée), ce qui permet à un *SafetyProvider* de mettre en œuvre simultanément plusieurs versions du présent document. Des *SafetyConsumers* de différentes versions peuvent ainsi accéder au même *SafetyProvider*.

6.2.5 DataTypes et longueur de SafetyData

Le présent document prend en charge l'envoi des *DataTypes Built-in* et *Simple* spécifiés dans la série IEC 62541 (voir l'IEC 62541-3 et l'IEC 62541-6), dans les *SafetyData*. Les *DataTypes* pris en charge sont spécifiques au fournisseur.

[RQ6.9] Seuls les *DataTypes* scalaires doivent être utilisés. Les matrices ne sont actuellement pas prises en charge par le présent document.

La longueur maximale prise en charge des *SafetyData* est spécifique au fournisseur, mais est toujours limitée à 1 500 octets. Les valeurs types pour la longueur maximale sont notamment 1 octet, 16 octets, 64 octets, 256 octets, 1 024 octets et 1 500 octets.

[RQ6.10] Pour les appareils de type contrôleur, les *DataTypes* pris en charge et la longueur maximale des *SafetyData* doivent être indiqués dans le manuel d'utilisation.

[RQ6.11] Pour le *DataType Boolean*, la valeur 0x01 doit être utilisée pour "true" et la valeur 0x00 doit être utilisée pour "false".

Il est recommandé d'envoyer plusieurs *Booleans* dans des variables distinctes. Cependant, dans les petits appareils, il peut être nécessaire de combiner un ensemble de 8 *Booleans* dans une *Variable* pour des questions de performances. Dans ce cas, le *DataType Byte* peut être utilisé.

6.2.6 Établissement de la connexion

Le présent document utilise les services de l'IEC 62541 pour l'établissement de la connexion et n'impose aucune exigence supplémentaire pour ces services.

La présente version du document décrit la configuration uniquement au stade de l'ingénierie. Cela signifie que les paramètres définis dans la *SPI* (voir 6.3.3.3 et 6.3.4.4) sont en lecture seule sur l'interface décrite dans le présent document. Il est prévu que la modification des paramètres soit effectuée selon une approche relative à la sécurité, en utilisant les outils et les interfaces respectifs fournis par le fournisseur. Les versions ultérieures du présent document peuvent spécifier une interface indépendante du fournisseur pour la configuration.

6.3 Interfaces de services

6.3.1 Vue d'ensemble

La Figure 7donne une vue d'ensemble de la *couche de communication de sécurité* et de ses interfaces. Elle représente ainsi également le domaine d'application du présent document. Le diagramme d'états qui gère le protocole constitue la fonction principale des services de couche. Les diagrammes d'états interagissent avec les interfaces suivantes:

– l'*interface du programme d'application de sécurité* (*SAPI*), à laquelle l'application de sécurité accède pour échanger des *SafetyData* pendant l'exécution;

– l'*interface de paramètres de sécurité* (*SPI*), accessible lors de la mise en service pour le réglage de paramètres de sécurité tels que les ID ou la valeur de temporisation dans le *SafetyConsumer*;

– l'*interface de diagnostic* (*DI*) *non relative à la sécurité*, à laquelle il est possible d'accéder au moment de l'exécution pour la résolution des problèmes de communication de sécurité;

– l'*interface de plateforme OPC UA* (*OPC UA PI*), qui connecte la *SCL* à la pile *non sécurisée* de l'IEC 62541 et qui est utilisée pendant l'exécution.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Les interfaces (*SAPI*, *SPI*, *DI* et *OPC UA PI*) décrites au 6.3 sont abstraites et informatives. Elles représentent les entrées et sorties de données logiques de cette couche qui sont nécessaires au bon fonctionnement du diagramme d'états. Aucun mapping normatif concret n'est spécifié. Les mises en œuvre concrètes sont spécifiques au fournisseur et ne correspondent pas nécessairement exactement aux interfaces abstraites décrites dans le présent document.

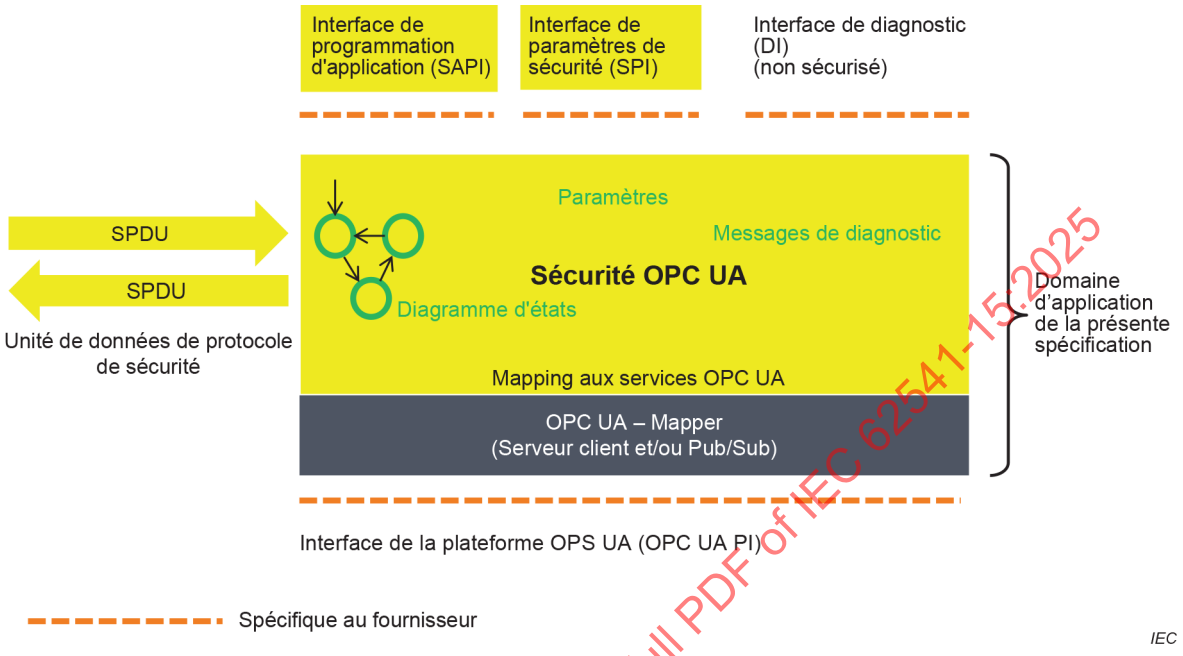


Figure 7 – Vue d'ensemble de la couche de communication de sécurité

6.3.2 Interface de plateforme OPC UA (OPC UA PI)

Les diagrammes d'états du présent document sont indépendants des services réels de l'IEC 62541 qui sont utilisés pour la transmission de données. Cela implique l'introduction d'un *mappeur OPC UA*, qui sert d'interface entre la *couche de communication de sécurité* et la pile de l'IEC 62541.

Le mappeur peut utiliser le *Client/Serveur* IEC 62541 et l'invocation de *Méthode* à distance, ou la publication de *Variables* distantes et l'abonnement à ces *Variables*, comme cela est défini dans l'IEC 62541-14. Les exigences relatives à la mise en œuvre du mappeur sont données implicitement au 6.2.

6.3.3 Interfaces du SafetyProvider

6.3.3.1 Généralités

La Figure 8 donne une vue d'ensemble des interfaces du *SafetyProvider*. La *SAPI* est spécifiée au 6.3.3.2 et la *SPI* est spécifiée au 6.3.3.3.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

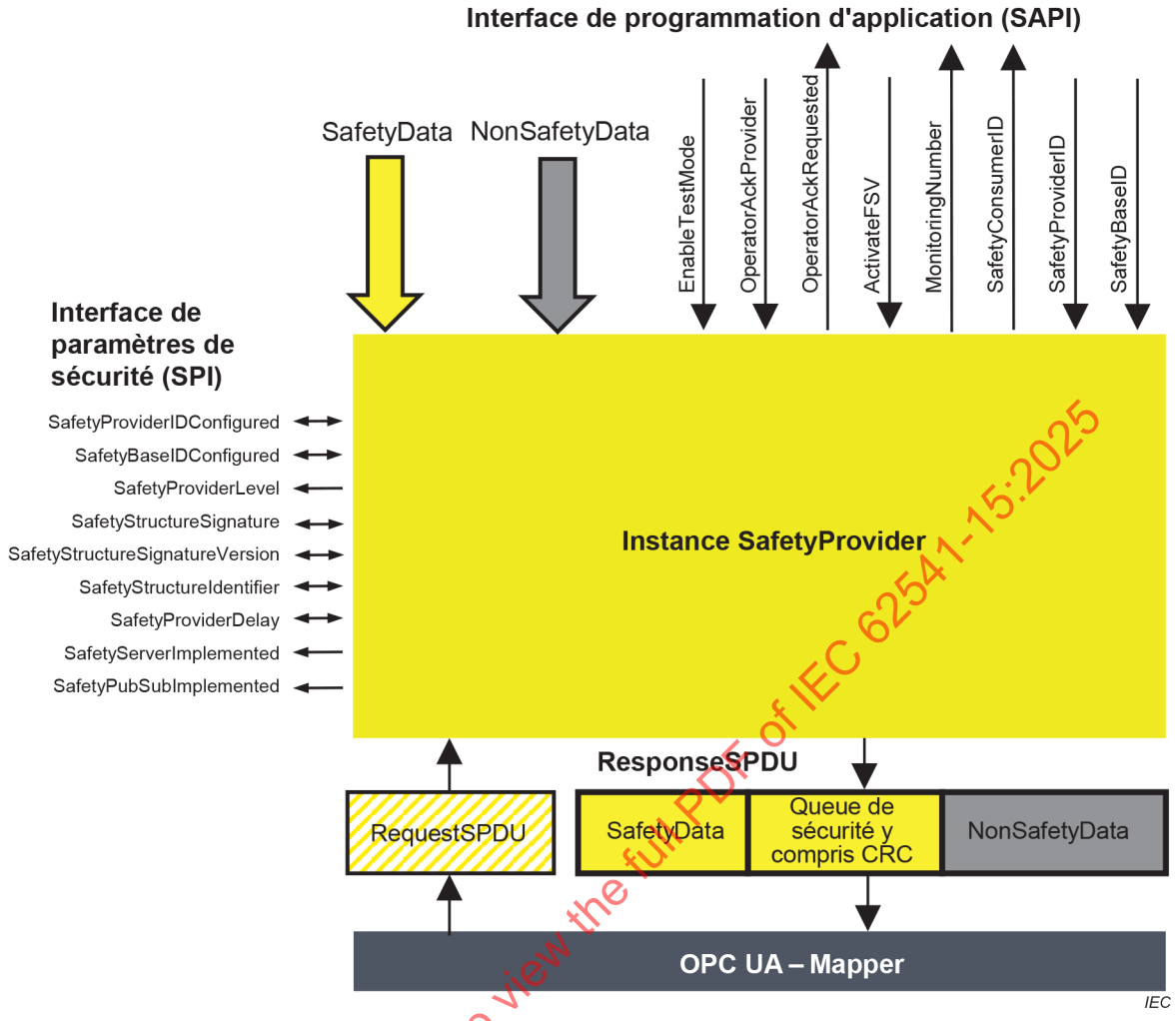


Figure 8 – Interfaces du SafetyProvider

6.3.3.2 SAPI du SafetyProvider

[RQ6.12] La *SAPI* du *SafetyProvider* représente les services de *couche de communication de sécurité* du *SafetyProvider*. Le Tableau 23 répertorie toutes les entrées et sorties de la *SAPI* du *SafetyProvider*. Chaque *SafetyProvider* doit mettre en œuvre la *SAPI* comme cela est indiqué dans le Tableau 23. Toutefois, les détails sont spécifiques au fournisseur.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Tableau 23 – SAPI du SafetyProvider

Terme de la SAPI	Type	I/O	Définition
SafetyData	Structure	I	Cette entrée est utilisée pour accepter les données utilisateur qui sont ensuite transmises sous la forme de *SafetyData* dans la *SPDU*. NOTE 1 Lorsqu'un nouveau *MNR* est reçu d'un *SafetyConsumer*, le diagramme d'états du *SafetyProvider* lit une nouvelle valeur des *SafetyData* provenant de son Application de sécurité correspondante et l'utilise jusqu'à réception du prochain *MNR*. Si aucune donnée utilisateur valide n'est disponible au niveau de l'Application de sécurité, ActivateFSV peut être défini sur "1" par l'Application de sécurité.
NonSafetyData	Structure	I	Utilisé pour transmettre de manière cohérente des valeurs de *NonSafetyData* (par exemple, des informations de diagnostic) avec des données de sécurité (voir 7.2.1.11).
EnableTestMode	Boolean	I	Lorsque cette entrée est définie sur "1", le *SafetyConsumer* **distant** est informé (par le bit 2 dans *ResponseSPDU.Flags*, voir 6.2.3.2) que les *SafetyData* sont des données d'essai et qu'elles ne sont pas destinées à être utilisées dans des décisions relatives à la sécurité. NOTE 2 Le présent document est destiné à être mis en œuvre exclusivement dans les appareils de sécurité (voir l'exigence RQ4.1).
OperatorAckProvider	Boolean	I	Cette entrée est utilisée pour mettre en œuvre un acquittement de l'opérateur du côté du *SafetyProvider*. La valeur est transférée au *SafetyConsumer*, où elle peut être utilisée pour déclencher le retour des valeurs de substitution *Failsafe* (*FSV*) aux *valeurs de processus* (*PV*) réelles (voir B.3.4).
OperatorAckRequested	Boolean	O	Indique qu'un acquittement de l'opérateur est demandé par le *SafetyConsumer*. Ce *fanion* est reçu dans la *RequestSPDU*.
ActivateFSV (valeurs de substitution *Failsafe*)	Boolean	I	Lorsque cette entrée est définie sur "1", le *SafetyConsumer* reçoit l'instruction (par l'intermédiaire du bit 1 de *ResponseSPDU.Flags*, voir 6.2.3.2) de fournir une *FSV* plutôt qu'une *PV* au programme d'application de sécurité. NOTE 3 S'il faut être en mesure de contrôler de manière plus précise le remplacement des *valeurs de processus* par des *FSV*, il est possible de le réaliser en utilisant des *qualificatifs* dans les *SafetyData* (voir 6.3.6).
SafetyConsumerID	UInt32	O	Cette sortie donne le ConsumerID utilisé lors du dernier accès à ce *SafetyProvider* par un *SafetyConsumer* (voir 6.2.2.3). Étant donné que toutes les vérifications relatives à la sécurité sont exécutées par une mise en œuvre du présent document, l'application de sécurité n'est pas tenue de vérifier ce *SafetyConsumerID*.
MonitoringNumber	UInt32	O	Cette sortie donne le *MonitoringNumber* (*MNR*). Elle est mise à jour chaque fois qu'une nouvelle demande est reçue du *SafetyConsumer*. Étant donné que toutes les vérifications relatives à la sécurité sont exécutées par une mise en œuvre du présent document, l'application de sécurité n'est pas tenue de vérifier ce *MonitoringNumber*.
SafetyProviderID	UInt32	I	Pour les systèmes dynamiques, cette entrée peut être définie sur une valeur différente de zéro. Dans ce cas, le *SafetyProvider* utilise cette valeur à la place de la valeur du paramètre *SafetyProviderIDConfigured* de la *SPI*. Si la valeur est remplacée par "0", la valeur du paramètre *SafetyProviderIDConfigured* de la *SPI* est (de nouveau) utilisée. Voir la Figure 8, le 3.1.2.15 et le 9.1.1. En règle générale, pour les systèmes statiques, cette entrée est toujours maintenue à la valeur "0".
SafetyBaseID	Guid	I	Pour les systèmes dynamiques, cette entrée peut être définie sur une valeur différente de zéro. Dans ce cas, le *SafetyProvider* utilise cette valeur à la place de la valeur du paramètre *SafetyBaseIDConfigured* de la *SPI*. Si la valeur est remplacée par "0", la valeur du paramètre *SafetyBaseIDConfigured* de la *SPI* est (de nouveau) utilisée. Voir la Figure 8, le 3.1.2.14 et le 9.1.1. En règle générale, pour les systèmes statiques, cette entrée est toujours maintenue à la valeur "0".

Pour plus d'informations sur les cas d'utilisation de l'acquittement de l'opérateur, voir l'Article B.3.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

6.3.3.3 SPI du SafetyProvider

[RQ6.13a] Chaque *SafetyProvider* doit mettre en œuvre les paramètres et les constantes [RQ6.13b], comme cela est indiqué dans le Tableau 24. Les paramètres (R/W dans la colonne "Accès") peuvent être définis par la *SPI*, tandis que les constantes (R dans la colonne "Accès") sont en lecture seule. Les mécanismes de réglage des paramètres sont spécifiques au fournisseur. La tentative de réglage d'un paramètre sur une valeur hors de sa plage ou le réglage d'un paramètre en lecture seule ne doivent pas être effectifs, et il convient qu'un message de diagnostic soit affiché le cas échéant. Les valeurs des constantes dépendent de la manière dont le *SafetyProvider* est mis en œuvre. Elles ne changent jamais et ne peuvent donc être écrites par aucune des interfaces.

Tableau 24 – SPI du SafetyProvider

Identificateur	Type	Plage	Valeur initiale (avant la configuration)	Accès	Note
SafetyProviderID Configured	UInt32	0x0 à 0xFFFFFFFF	0x0	R/W	*SafetyProviderID* du *SafetyProvider* normalement utilisé (voir 3.1.2.13, 3.1.2.15 et 9.1.1). Pour les systèmes dynamiques, le programme d'application de sécurité peut écraser cet ID en fournissant une valeur différente de zéro au *SafetyProviderID* d'entrée de la *SAPI* du *SafetyProvider*. Cette valeur d'exécution peut être interrogée à l'aide du paramètre *SafetyProviderIDActive*. Voir le 6.2.2.6 pour des détails sur les valeurs configurées et actives. NOTE 1 Si les valeurs fournies à la *SPI* et à la *SAPI* sont 0x0, cela signifie que le *SafetyProvider* n'est pas correctement configuré. Les *SafetyConsumers* ne tentent jamais de communiquer avec les *SafetyProviders* ayant un *SafetyProviderID* de 0x0 (voir les Transitions T13/T27 dans le Tableau 35 et la macro <ParametersOK?> dans le Tableau 33).
SafetyBaseID-Configured	Guid	Toute valeur qui peut être repré-sentée avec seize octets	Les seize octets sont définis à 0x00	R/W	*SafetyBaseID* du *SafetyProvider* normalement utilisé (voir 3.1.2.14, 3.1.2.13 et 9.1.1). Pour les systèmes dynamiques, le programme d'application de sécurité peut écraser cet ID en fournissant une valeur différente de zéro au *SafetyBaseID* d'entrée de la *SAPI* du *SafetyProvider*. Cette valeur d'exécution peut être interrogée à l'aide du paramètre *SafetyBaseIDActive*. Voir le 6.2.2.6 pour des détails sur les valeurs configurées et actives. NOTE 2 Si les valeurs fournies à la *SPI* et à la *SAPI* sont 0x0, cela signifie que le *SafetyProvider* n'est pas correctement configuré. Les *SafetyConsumers* ne tentent jamais de communiquer avec les *SafetyProviders* ayant un *SafetyBaseID* de 0x0 (voir les Transitions T13/T27 dans le Tableau 35 et la macro <ParametersOK?> dans le Tableau 33). Voir le 9.1.1 pour plus d'informations sur le *GUID*.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Identificateur	Type	Plage	Valeur initiale (avant la configuration)	Accès	Note
SafetyProvider-Level	Byte	0x01 à 0x04	n.a.	R	*SIL* qu'est capable de prendre en charge la mise en œuvre du *SafetyProvider* (matériel et logiciel) (voir Figure 9). NOTE 3 Il est indépendant de la génération des *SafetyData* au niveau de la *SAPI*. NOTE 4 Le *SafetyProviderLevel* est utilisé pour distinguer les appareils ayant un *SIL* différent. Par conséquent, les *SPDU* provenant d'un appareil avec un *SIL* faible ne sont jamais acceptées lorsqu'un *SafetyConsumer* est paramétré pour mettre en œuvre une fonction de sécurité avec un *SIL* élevé.
SafetyStructure Signature	UInt32	0x0 à 0xFFFFFFFF	0x0	R/W	Signature de la structure des *SafetyData* pour le calcul (voir 7.2.3.5). NOTE 5 "0" ne serait pas une signature valide et indique donc un *SafetyProvider* qui n'est pas correctement configuré. Les *SafetyConsumers* ne tentent jamais de communiquer avec les *SafetyProviders* ayant une *SafetyStructureSignature* de 0x0 (voir les Transitions T13/T27 dans le Tableau 35 et la macro <ParametersOK?> dans le Tableau 33).
SafetyStructure-SignatureVersion	UInt16	0x1	0x1	R/W	Version utilisée pour calculer la *SafetyStructureSignature* (voir 7.2.3.5).
SafetyStructure-Identifier	String	toutes les chaînes	"" (chaîne vide)	R/W	Identificateur qui décrit le *DataType* des *SafetyData* (voir 7.2.3.5).
SafetyProvider-Delay	UInt32	0x0 à 0xFFFFFFFF	0x0	R/W	En microsecondes (µs). Peut être réglé pendant la phase d'ingénierie du *SafetyProvider* ou lors de la configuration en ligne. *SafetyProviderDelay* est le temps maximal que met le *SafetyProvider* entre la réception de la *RequestSPDU* et le début de la transmission de la *ResponseSPDU* (voir 8.1). Le paramètre *SafetyProviderDelay* n'a aucune influence sur le comportement fonctionnel du *SafetyProvider*. Par conséquent, il n'est pas nécessaire de générer cette valeur selon une approche relative à la sécurité. Elle est toutefois fournie dans le *Modèle d'information* IEC 62541 d'un *SafetyProvider* pour informer de son délai le plus défavorable. La valeur peut être utilisée lors de la mise en service pour vérifier si le comportement de temporisation du *SafetyProvider* est approprié pour respecter le délai de fonctionnement du chien de garde du *SafetyConsumer* correspondant.
SafetyServer-Implemented	Boolean	0x0 ou 0x1	n.a.	R	Ce paramètre en lecture seule indique si le *SafetyProvider* a mis en œuvre la partie *Serveur* de la communication *Client/Serveur* IEC 62541 (voir 5.4): 1: le *Serveur* pour la communication *Client/Serveur* IEC 62541 est mis en œuvre; 0: le *Serveur* pour la communication *Client/Serveur* IEC 62541 n'est pas mis en œuvre. Les *Facettes* correspondantes sont *SafetyProviderServer* et *SafetyProviderServer-Mapper*.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Identificateur	Type	Plage	Valeur initiale (avant la configuration)	Accès	Note
SafetyPubSub-Implemented	Boolean	0x0 ou 0x1	n.a.	R	Ce paramètre en lecture seule indique si le *SafetyProvider* a mis en œuvre les éditeurs et les abonnés nécessaires pour la communication *PubSub* IEC 62541 (voir 5.4): 1: la communication *PubSub* IEC 62541 est mise en œuvre; 0: la communication *PubSub* IEC 62541 n'est pas mise en œuvre. Les *Facettes* correspondantes sont *SafetyProviderPubSub* et *SafetyProviderPubSubMapper*.

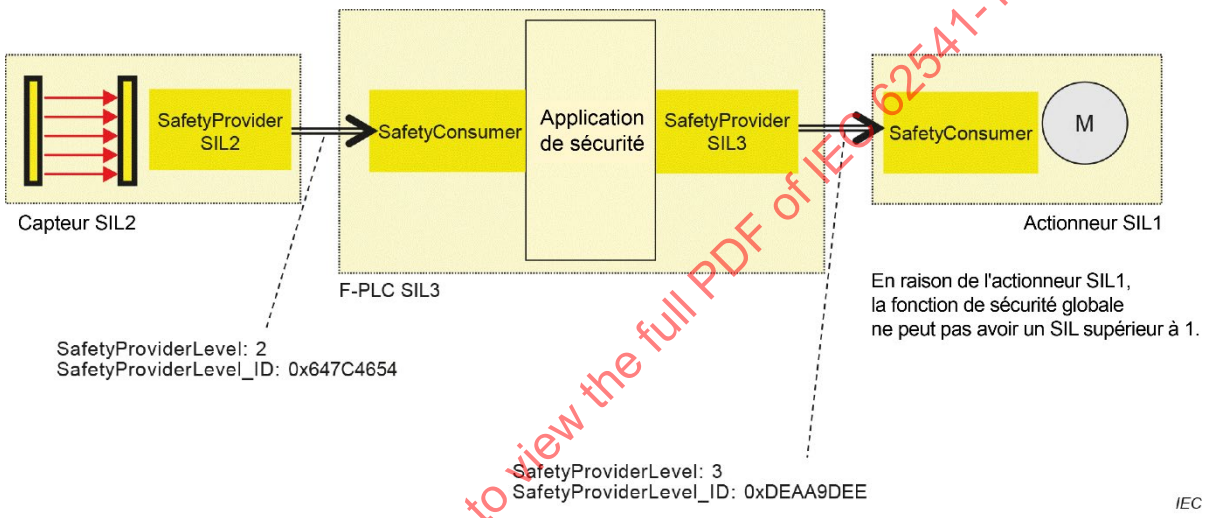


Figure 9 – Exemples de combinaisons de capacités SIL

La constante *SafetyProviderLevel* détermine la valeur utilisée pour *SafetyProviderLevel_ID* lors du calcul du *SPDU_ID* (voir 7.2.3.3).

NOTE *SafetyProviderLevel* est défini comme le *SIL* qu'est capable de prendre en charge la mise en œuvre du *SafetyProvider* (matériel et logiciel), à ne pas confondre avec le *SIL* de la fonction de sécurité mise en œuvre. Par exemple, la Figure 9 représente une fonction de sécurité qui est mise en œuvre à l'aide d'un capteur capable de prendre en charge le niveau *SIL*2, d'un PLC capable de prendre en charge le niveau *SIL*3 et d'un actionneur capable de prendre en charge le niveau *SIL*1. Le *SIL* global de la fonction de sécurité est assimilé au *SIL*1. Néanmoins, le *SafetyProvider* mis en œuvre sur le capteur utilise la valeur constante "2" comme *SafetyProviderLevel*, tandis que le *SafetyProvider* mis en œuvre sur le PLC utilise la valeur constante "3" comme *SafetyProviderLevel*.

Il est nécessaire que les *SafetyConsumers* correspondants (sur le PLC et l'actionneur) connaissent le *SafetyProviderLevel* de leurs *SafetyProviders* pour pouvoir vérifier le *SPDU_ID* (voir 7.2.3.2).

6.3.4 Interfaces du SafetyConsumer

6.3.4.1 Généralités

La Figure 10 donne une vue d'ensemble des interfaces du *SafetyConsumer*. L'*interface du programme d'application de sécurité* (*SAPI*) est spécifiée au 6.3.4.2 et l'*interface de paramètres de sécurité* (*SPI*) est spécifiée au 6.3.4.4.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

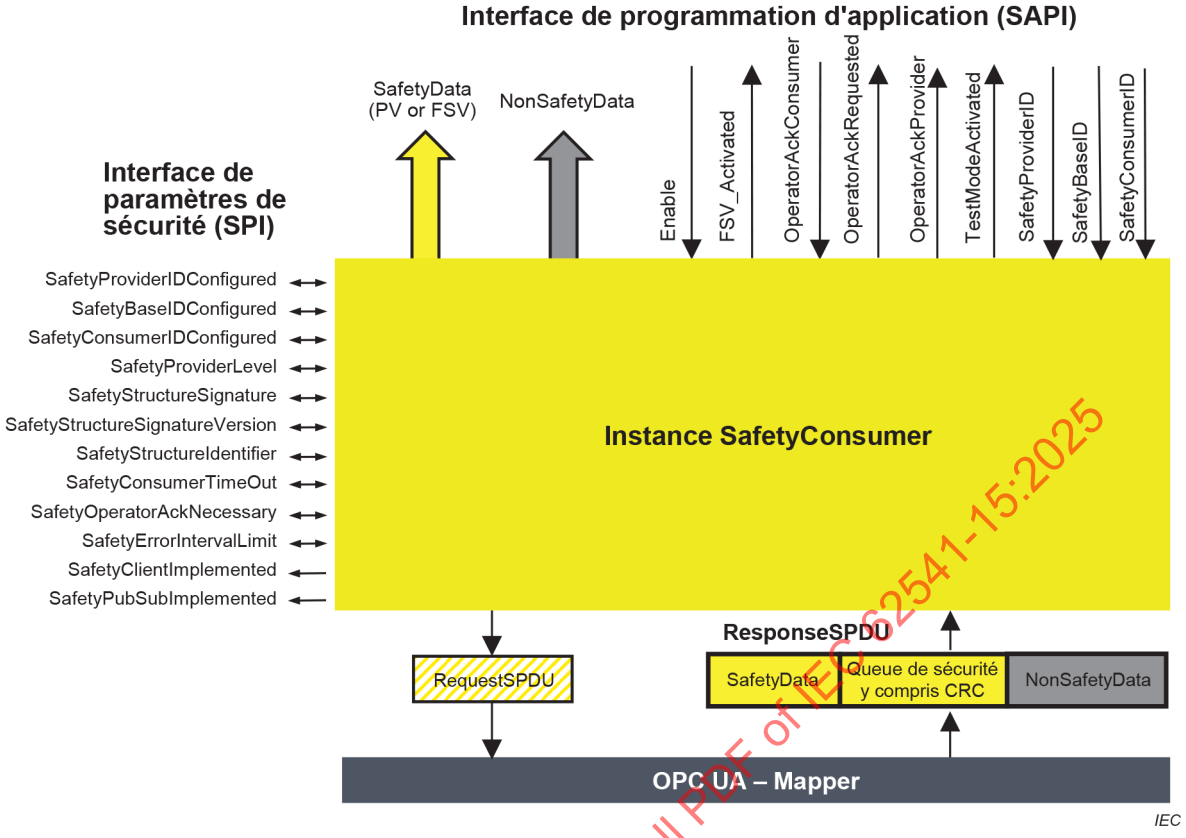


Figure 10 – Interfaces du SafetyConsumer

6.3.4.2 SAPI du SafetyConsumer

La *SAPI* du *SafetyConsumer* représente les services de *couche de communication de sécurité* du *SafetyConsumer*. Le Tableau 25 répertorie toutes les entrées et sorties de la *SAPI* du *SafetyConsumer*.

[RQ6.14] Chaque *SafetyConsumer* doit mettre en œuvre la *SAPI* comme cela est indiqué dans le Tableau 25. Toutefois, les détails sont spécifiques au fournisseur.

Tableau 25 – SAPI du SafetyConsumer

Terme de la SAPI	Type	I/O	Définition
SafetyData	Structure	O	Cette sortie fournit soit les *valeurs de processus* reçues du *SafetyProvider* dans le champ *SafetyData* de la *SPDU*, soit les *FSV*.
NonSafetyData	Structure	O	Cette sortie fournit les *valeurs de processus non relatives à la sécurité* (par exemple, informations de diagnostic) qui ont été envoyées avec des données de sécurité (voir 7.2.1.11).
Enable	Boolean	I	Lorsque cette entrée est remplacée par "0", le *SafetyConsumer* remplace chaque *Variable* des *SafetyData* par "0"[1] et arrête d'envoyer des demandes au *SafetyProvider*. Lorsque *Enable* est défini sur "1", le *SafetyConsumer* relance la communication sûre. La variable peut être utilisée pour retarder le début de la communication de sécurité conformément au présent document, après la mise sous tension, jusqu'à ce que "connexion IEC 62541 prête" soit défini. Le délai n'est pas surveillé lorsque Enable est défini sur "0".
FSV_Activated	Boolean	O	Cette sortie indique par "1" que les *FSV* des *SafetyData* (toutes les valeurs binaires "0") sont fournies à la sortie[1]. NOTE 1 Si la vérification de la *ResponseSPDU* indique une erreur, FSV_Activated est défini sur "1" (voir T24 dans le Tableau 35). Il peut y avoir d'autres raisons.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Terme de la SAPI	Type	I/O	Définition
OperatorAck-Consumer	Boolean	I	Pour la motivation, voir le 6.3.4.3. Après une indication de OperatorAckRequested, cette entrée peut être utilisée pour signaler un acquittement de l'opérateur. Lorsque cette entrée est modifiée pour passer de "0" à "1" (front montant), le *SafetyConsumer* reçoit l'instruction de remplacer les *FSV* des *SafetyData* par des *PV*. OperatorAckConsumer est traité uniquement si ce front montant arrive après la définition du paramètre OperatorAckRequested sur "1" (voir Figure 19). Si un front montant du paramètre OperatorAckConsumer arrive avant le passage d'OperatorAckRequested à 1, ce front montant est ignoré.
OperatorAck-Requested	Boolean	O	Cette sortie indique la demande d'acquittement de l'opérateur. Le bit est défini sur "1" par le *SafetyConsumer* lorsque les trois conditions suivantes sont remplies: 1) un trop grand nombre d'erreurs de communication avant été détectées par le passé, le *SafetyConsumer* a décidé de passer à des valeurs de substitution *Failsafe*; 2) aucune erreur de communication ne se produit actuellement donc l'acquittement de l'opérateur est possible; 3) l'acquittement de l'opérateur (front montant au niveau de l'OperatorAckConsumer d'entrée) ne s'est pas encore produit. Le bit est remis à "0" lorsqu'un front montant au niveau de l'OperatorAckConsumer est détecté.
OperatorAck-Provider	Boolean	O	Cette sortie indique qu'un acquittement de l'opérateur a été effectué sur le *SafetyProvider*. S'il convient d'autoriser l'acquittement de l'opérateur au niveau du *SafetyProvider*, cette sortie est connectée à OperatorAckConsumer (voir B.3.4 et B.3.5). NOTE 2 Si la vérification de la *ResponseSPDU* indique une erreur, cette sortie reste à sa dernière valeur (voir T24 dans le Tableau 35).
TestModeActivated	Boolean	O	Il est attendu que le programme d'application de sécurité évalue cette sortie pour déterminer si le partenaire de communication est en mode d'essai ou non. La valeur "1" indique que le partenaire de communication (source de données) est en mode d'essai, par exemple lors de la mise en service. Les données provenant d'un appareil en mode d'essai peuvent être utilisées pour les essais, mais elles ne sont pas destinées à être utilisées pour contrôler les processus essentiels à la sécurité. La valeur "0" représente le mode "normal" relatif à la sécurité. Le mode d'essai permet au programmeur et à la personne chargée de la mise en service de valider l'application de sécurité à l'aide de données d'essai. NOTE 3 Si la vérification de la *ResponseSPDU* indique une erreur et si le *ErrorIntervalTimer* (voir 6.3.4.4) n'a pas également expiré, TestModeActivated est réinitialisé (voir l'état S17_Error dans le Tableau 34.
SafetyProviderID	UInt32	I	Pour les systèmes dynamiques, cette entrée peut être définie sur une valeur différente de zéro. Dans ce cas, le *SafetyConsumer* utilise cette variable au lieu du paramètre *SafetyProviderIDConfigured* de la *SPI*. Cette entrée est uniquement lue au cours du premier cycle, ou lorsqu'un front montant se produit au niveau de l'entrée Enable. Voir également le Tableau 26. Si cette valeur est remplacée par "0", la valeur du paramètre *SafetyProviderIDConfigured* de la *SPI* est (de nouveau) utilisée. En règle générale, pour les systèmes statiques, cette entrée est toujours maintenue à la valeur "0".
SafetyBaseID	Guid	I	Pour les systèmes dynamiques, cette entrée peut être définie sur une valeur différente de zéro. Dans ce cas, le *SafetyConsumer* utilise cette variable au lieu du paramètre *SafetyBaseIDConfigured* de la *SPI*. Cette entrée est uniquement lue au cours du premier cycle, ou lorsqu'un front montant se produit au niveau de l'entrée Enable. Voir également le Tableau 26. Si cette valeur est remplacée par "0", le paramètre *SafetyBaseIDConfigured* de la *SPI* est activé. En règle générale, pour les systèmes statiques, cette entrée est toujours maintenue à la valeur "0".

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Terme de la SAPI	Type	I/O	Définition
SafetyConsumerID	UInt32	I	Pour les systèmes dynamiques, cette entrée peut être définie sur une valeur différente de zéro. Dans ce cas, le *SafetyConsumer* utilise cette variable au lieu du paramètre *SafetyConsumerID* de la *SPI*. Cette entrée est uniquement lue au cours du premier cycle, ou lorsqu'un front montant se produit au niveau de l'entrée Enable. Voir également le Tableau 26. Si cette valeur est remplacée par "0", le paramètre *SafetyConsumerID* de la *SPI* est activé. En règle générale, pour les systèmes statiques, cette entrée est toujours maintenue à la valeur "0".
[1] Si une application exige une *FSV* différente de "toutes les valeurs binaires 0", elle est supposée utiliser les constantes appropriées et ignorer la sortie de *SafetyData* lorsque FSV_Activated est défini.			

Pour plus d'informations sur les cas d'utilisation de l'acquittement de l'opérateur, voir l'Article B.3.

6.3.4.3 Motivation pour l'acquittement de l'opérateur sur la SAPI (OperatorAckConsumer)

L'argumentation de sécurité part du principe que les erreurs aléatoires dans la pile sous-jacente de l'IEC 62541, y compris ses liaisons de communication, ne sont pas très fréquentes, c'est-à-dire que son taux de défaillance est inférieur à un seuil donné, en fonction du *SIL* souhaité (voir 9.3.1).

Chaque fois que le *SafetyConsumer* détecte un message erroné, il vérifie si l'hypothèse est toujours valide et passe à des valeurs de substitution *Failsafe* si ce n'est pas le cas. Le retour à des *valeurs de processus* exige alors un acquittement de l'opérateur.

L'acquittement de l'opérateur est présumé initié par un opérateur humain chargé de vérifier l'installation (voir Tableau 40, ligne "Acquittement de l'opérateur"). C'est pour cette raison que le *SafetyConsumer* fournit le paramètre OperatorAckRequested à l'application de sécurité. Voir l'Article B.2 pour plus d'informations sur les scénarios d'acquittement de l'opérateur.

Les erreurs de temporisation exigent un acquittement de l'opérateur seulement si l'acquittement de l'opérateur est exigé par la fonction de sécurité elle-même. Dans ce cas, SafetyOperatorAckNecessary est défini pour indiquer que des acquittements de l'opérateur sont exigés. Voir le 6.3.4.5 pour plus d'informations.

6.3.4.4 SPI du SafetyConsumer

[RQ6.15a] Chaque *SafetyConsumer* doit mettre en œuvre les paramètres et les constantes [RQ6.15b], comme cela est indiqué dans le Tableau 26. Les paramètres (R/W dans la colonne "Accès") peuvent être définis par la *SPI*, tandis que les constantes (R dans la colonne "Accès") sont en lecture seule. Les mécanismes de réglage de ces paramètres sont spécifiques au fournisseur. La tentative de réglage d'un paramètre sur une valeur hors de sa plage ou le réglage d'un paramètre en lecture seule ne doivent pas être effectifs, et il convient qu'un message de diagnostic soit affiché le cas échéant. La *SPI* du *SafetyConsumer* représente les paramètres de gestion de la *couche de communication de sécurité* du *SafetyConsumer*. Les valeurs des constantes dépendent de la manière dont le *SafetyConsumer* est mis en œuvre. Elles ne changent jamais et ne peuvent donc être écrites par aucune des interfaces.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Tableau 26 – SPI du SafetyConsumer

Identificateur	Type	Plage valide	Valeur initiale (avant la configuration)	Accès	Note
SafetyProvider-IDConfigured	UInt32	0x0 à 0xFFFFFFFF	0x0	R/W	*SafetyProviderID* par défaut du *SafetyProvider* que ce *SafetyConsumer* utilise pour établir une connexion (voir Figure 8 et 3.1.2.15). Pour les systèmes dynamiques, le programme d'application de sécurité peut écraser cet ID en fournissant une valeur différente de zéro au *SafetyProviderID* d'entrée de la *SAPI* du *SafetyConsumer*. Cette valeur d'exécution peut être interrogée à l'aide du paramètre *SafetyProviderIDActive*. Voir le 6.2.2.6 pour des détails sur les valeurs configurées et actives.
SafetyBaseID-Configured	Guid	Toute valeur qui peut être repré-sentée avec seize octets.	Les seize octets sont définis à 0x0.	R/W	*SafetyBaseID* par défaut du *SafetyProvider* que ce *SafetyConsumer* utilise pour établir une connexion (voir 3.1.2.14). Pour les systèmes dynamiques, le programme d'application de sécurité peut écraser cet ID en fournissant une valeur différente de zéro au *SafetyBaseID* d'entrée de la *SAPI* du *SafetyConsumer*. Cette valeur d'exécution peut être interrogée à l'aide du paramètre *SafetyBaseIDActive*. Voir le 6.2.2.6 pour des détails sur les valeurs configurées et actives. Voir le 9.1.1 pour plus d'informations sur le *GUID*.
SafetyConsumer IDConfigured	UInt32	0x0 à 0xFFFFFFFF	0x0	R/W	*SafetyConsumerID* du *SafetyConsumer* (voir 9.1.2). Pour les systèmes dynamiques, le programme d'application de sécurité peut écraser cet ID en fournissant une valeur différente de zéro au *SafetyConsumerID* d'entrée de la *SAPI* du *SafetyConsumer*. Cette valeur d'exécution peut être interrogée à l'aide du paramètre *SafetyConsumerIDActive*. Voir le 6.2.2.6 pour des détails sur les valeurs configurées et actives.
SafetyProvider-Level	Byte	0x01 à 0x04	0x04	R/W	Attente du *SafetyConsumer* quant au *SIL* qu'est capable de prendre en charge la mise en œuvre du *SafetyProvider* (matériel et logiciel). Voir le 7.2.3.3 et la Figure 9.
SafetyStructure-Signature	UInt32	0x0 à 0xFFFFFFFF	0x0	R/W	Signature sur la structure des *SafetyData* (voir 7.2.3.5).
SafetyStructure-SignatureVersion	UInt16	0x1	0x1	R/W	Version utilisée pour calculer la *SafetyStructureSignature* (voir 7.2.3.5). Pour le *SafetyConsumer*, ce paramètre est facultatif.
SafetyStructure-Identifier	String		""	R/W	Identificateur qui décrit le *DataType* des *SafetyData* (voir 7.2.3.5). Pour le *SafetyConsumer*, ce paramètre est facultatif.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Identificateur	Type	Plage valide	Valeur initiale (avant la configuration)	Accès	Note
SafetyConsumer Timeout	UInt32	0x0 à 0xFFFFFFFF	0x0	R/W	Temps de fonctionnement du chien de garde en microsecondes (µs). Lorsque le *SafetyConsumer* envoie une demande à un *SafetyProvider*, le temporisateur de son chien de garde est défini sur cette valeur. L'expiration de ce temporisateur avant la réception d'une réponse sans erreur par le *SafetyProvider* indique un retard inacceptable. Voir le 8.1
SafetyOperator-AckNecessary	Boolean	0x0 ou 0x1	0x1	R/W	Ce paramètre permet de déterminer si un acquittement de l'opérateur (OA) est nécessaire en cas d'erreurs de type "retard inacceptable" ou "perte", ou lorsque le *SafetyProvider* a activé les *FSV* (ActivateFSV). 1: les *FSV* sont fournies au niveau des *SafetyData* de sortie de la *SAPI* jusqu'à l'OA; 0: les *PV* sont fournies au niveau des *SafetyData* de la *SAPI* dès que la communication est sans erreur. Dans le cas d'ActivateFSV, les valeurs passent de *FSV* à *PV* dès que le paramètre ActivateFSV revient à "0". NOTE Ce paramètre n'a pas d'influence sur le comportement du *SafetyConsumer* après la détection d'autres types d'erreurs de communication, telles qu'une corruption des données ou une erreur détectée par le *SPDU_ID*. Pour ces types d'erreurs, un OA est obligatoire (voir 6.3.4.3).
SafetyError-IntervalLimit	UInt16	6, 60, 600	600	R/W	Valeur en minutes. Le paramètre *SafetyErrorIntervalLimit* détermine l'intervalle de temps minimal entre deux erreurs de communication consécutives, de sorte qu'elles ne déclenchent pas le passage à des *FSV* dans le *SafetyConsumer* (voir 6.3.4.3). Il a une influence sur la disponibilité et sur la valeur PFH et/ou PFDavg de la liaison de communication de sécurité conforme au présent document (voir 9.4).
SafetyClient-Implemented	Boolean	0x0 ou 0x1	n.a.	R	Ce paramètre en lecture seule indique si le *SafetyConsumer* a mis en œuvre la partie client de la communication *Client/Serveur* IEC 62541 (voir 5.4): 1: le Client pour la communication *Client/Serveur* IEC 62541 est mis en œuvre; 0: le Client pour la communication *Client/Serveur* IEC 62541 n'est pas mis en œuvre. La *Facette* correspondante est *SafetyConsumerClient*.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Identificateur	Type	Plage valide	Valeur initiale (avant la configuration)	Accès	Note
SafetyPubSub-Implemented	Boolean	0x0 ou 0x1	n.a.	R	Ce paramètre en lecture seule indique si le *SafetyConsumer* a mis en œuvre les éditeurs et les abonnés nécessaires pour la communication *PubSub* IEC 62541 (voir 5.4): 1: la communication *PubSub* IEC 62541 est mise en œuvre; 0: la communication *PubSub* IEC 62541 n'est pas mise en œuvre. Les *Facettes* correspondantes sont *SafetyConsumerPubSub* et *SafetyConsumerPubSubMapper*.

6.3.4.5 Motivation pour le paramètre SafetyOperatorAckNecessary de la SPI

Ce paramètre détermine si un redémarrage automatique (c'est-à-dire un rebasculement automatique des valeurs *Failsafe* sur des *valeurs de processus*) est possible ou non pour la fonction de sécurité. Il est attendu que ce paramètre soit réglé à 1 pour les fonctions de sécurité qui autorisent un redémarrage automatique et où le redémarrage exige toujours une interaction humaine.

Si le redémarrage automatique de la fonction de sécurité est sûr, le paramètre peut être mis à 0.

6.3.5 Communication de sécurité cyclique et acyclique

Le présent document prend en charge la communication de sécurité cyclique et acyclique.

Pour la plupart des fonctions de sécurité, il est nécessaire de réagir dans les meilleurs délais aux événements externes, par exemple l'activation d'un bouton d'arrêt d'urgence ou la coupure d'une barrière immatérielle. Dans ces applications, une communication de sécurité cyclique est établie. Cela signifie que le *SafetyConsumer* est exécuté de manière cyclique, et que le temps entre deux exécutions consécutives est limité de manière sûre. Le temps maximal entre deux exécutions du *SafetyConsumer* contribue au *temps de réponse de la fonction de sécurité* (*SFRT*).

Certaines fonctions de sécurité, comme le transfert des données de configuration de sécurité au démarrage, n'ont pas à réagir aux événements externes. Dans ce cas, il n'est pas exigé d'exécuter le *SafetyConsumer* de manière cyclique.

6.3.6 Principe applicable aux "variables d'application avec qualificatif"

Les *qualificatifs* permettent au *SafetyProvider* d'indiquer l'exactitude des valeurs à un niveau précis. Il est judicieux d'associer des *qualificatifs* à chaque valeur individuelle envoyée dans une *SPDU*. Les *qualificatifs* font partie des *SafetyData* et ne relèvent donc pas du domaine d'application du présent document.

[RQ6.16] Toutefois, lorsque des *qualificatifs* sont utilisés, les valeurs indiquées dans le Tableau 27 doivent être utilisées, c'est-à-dire 0x1 pour une valeur valide ("good") et 0x0 pour une valeur non valide ("bad").

Tableau 27 – Exemples de "variables d'application avec qualificatif"

Valeur	Qualificatif
valide	0x1 (= good)
non valide	0x0 (= bad)

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

La vérification des *qualificatifs* est effectuée dans l'application de sécurité.

6.4 Diagnostics

6.4.1 Généralités

Les diagnostics conformes au présent document peuvent être mis en œuvre selon une approche *non relative à la sécurité*. Cela permet de catégoriser et de localiser les erreurs de communication de sécurité.

Le présent document fournit deux types de diagnostics:

– les messages de diagnostic générés par le *SafetyConsumer* et fournis d'une manière spécifique au fournisseur;
– la *Méthode ReadSafetyDiagnostics*, définie dans le *Modèle d'information* IEC 62541 (voir 6.2.2.4 et 6.4.3).

6.4.2 Messages de diagnostic du SafetyConsumer

[RQ6.17] Chaque fois que la macro <Set Diag(SD_IDerroA, isPermanent)> est exécutée dans le *SafetyConsumer*, la représentation textuelle indiquée dans le Tableau 28 doit être présentée. Les détails et l'emplacement de cette représentation (affichage, fichier journal, etc.) sont spécifiques au fournisseur.

Tableau 28 – Messages de diagnostic de la couche de sécurité

Identificateur interne (utilisé dans les diagrammes d'états)	Type d'erreur générale (chaîne)	Type d'erreur étendue (chaîne)	Code d'erreur (décalage)[1]	Classification *) (facultatif)	Obligatoire
SD_IDerrIgn	Le *SafetyConsumer* a rejeté un message en raison d'un ID incorrect.		0x01	A	Oui
SD_IDerrOA	Le *SafetyConsumer* est passé à des *valeurs de substitution Failsafe* en raison d'un ID incorrect. Un acquittement de l'opérateur est exigé.	Discordance de *SafetyBaseID*.[2]	0x11	B, E	Oui
SD_IDerrOA	Le *SafetyConsumer* est passé à des *valeurs de substitution Failsafe* en raison d'un ID incorrect. Un acquittement de l'opérateur est exigé.	Discordance de *SafetyProviderID*.	0x12	B, E	Oui
SD_IDerrOA	Le *SafetyConsumer* est passé à des *valeurs de substitution Failsafe* en raison d'un ID incorrect. Un acquittement de l'opérateur est exigé.	Discordance de la structure ou de l'identificateur des *SafetyData*.[3]	0x13	B, E	Oui
SD_IDerrOA	Le *SafetyConsumer* est passé à des *valeurs de substitution Failsafe* en raison d'un ID incorrect. Un acquittement de l'opérateur est exigé.	Discordance de *SafetyProvider-Level*.[4]	0x14	B, E	Oui
CRCerrIgn	Le *SafetyConsumer* a rejeté un message en raison d'une erreur de *CRC* (corruption de données).		0x05	A	Oui
CRCerrOA	Le *SafetyConsumer* est passé à des *valeurs de substitution Failsafe* en raison d'une erreur de *CRC* (corruption de données). Un acquittement de l'opérateur est exigé.		0x15	B, C	Oui

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Identificateur interne (utilisé dans les diagrammes d'états)	Type d'erreur générale (chaîne)	Type d'erreur étendue (chaîne)	Code d'erreur (décalage)[1]	Classification *) (facultatif)	Obligatoire
ColDerrIgn	Le *SafetyConsumer* a rejeté un message en raison d'un ConsumerID incorrect.		0x06	A	Oui
ColDerrOA	Le *SafetyConsumer* est passé à des *valeurs de substitution Failsafe* en raison d'un SafetyConsumerID incorrect. Un acquittement de l'opérateur est exigé.		0x16	B	Oui
MNRerrIgn	Le *SafetyConsumer* a rejeté un message en raison d'un *MonitoringNumber* incorrect.		0x07	A	Oui
MNRerrOA	Le *SafetyConsumer* est passé à des *valeurs de substitution Failsafe* en raison d'un numéro de surveillance incorrect. Un acquittement de l'opérateur est exigé.		0x17	B, C	Oui
CommErrTO	Le *SafetyConsumer* est passé à des *valeurs de substitution Failsafe* en raison d'une temporisation.		0x08	B	Oui
ApplErrTO	Le *SafetyConsumer* est passé à des *valeurs de substitution Failsafe* à la demande de l'application de sécurité.		0x09	D	Non
ParametersInvalid	Le *SafetyConsumer* a été configuré avec des paramètres non valides.		0x0A	B, E	Oui
FSV_Requested	Le *SafetyConsumer* est passé à des *valeurs de substitution Failsafe* à la demande du *SafetyProvider*. Un acquittement de l'opérateur est exigé.[5]		0x20	F	Oui

[1] Un décalage de 0x10 ou plus indique une erreur qui exige l'acquittement de l'opérateur.

[2] Ce texte peut également être affiché lorsque l'erreur dans le *SPDU_ID* est due à un *SafetyBaseID* incorrect.

[3] Ce texte peut également être affiché lorsque l'erreur dans le *SPDU_ID* est due à un *SafetyStructureID* incorrect.

[4] Ce texte peut également être affiché lorsque l'erreur dans le *SPDU_ID* est due à un *SafetyProviderLevel* incorrect.

[5] Un message de diagnostic est généré uniquement si le paramètre *SPI.SafetyOperatorAckNecessary* est défini sur "true" (voir la Transition T22 dans le Tableau 35).

*) La classification suivante est spécifiée:

A) Erreur de communication transitoire

B) Erreur de communication permanente

C) La qualité de transmission ne semble pas suffisante

D) Erreur d'application

E) Erreur de paramètre

F) L'erreur n'a aucun effet sur la communication elle-même.

Afin d'éviter un débordement de messages de diagnostic en cas d'erreurs de transmission, seuls deux messages au maximum sont présentés même si plusieurs erreurs de communication se produisent successivement. Cela est assuré par le comportement défini dans le diagramme d'états du *SafetyConsumer*.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Fonctionnalités facultatives (spécifiques au fournisseur):

- Élargir les données de diagnostic par la valeur attendue et la valeur reçue, par exemple: discordance de *SafetyProviderID*:
  SafetyProviderID attendu 0x00000005
  SafetyProviderID reçu: 0x00000007
- Élargir les données de diagnostic si un paramètre du *SafetyConsumer* n'est pas valide.
  Exemple 1:
  Le *SafetyConsumer* a été configuré avec des paramètres non valides.
  La valeur 0x00000000 est un *SafetyProviderID* non valide.

6.4.3 Méthode ReadSafetyDiagnostics du SafetyProvider

Cette *Méthode* (dans le cadre du *mappeur OPC UA*) sert d'*interface de diagnostic* et est présente pour chaque *SafetyProvider*. Pour l'observation de séries chronologiques, cette interface peut être interrogée, par exemple par un appareil de diagnostic. Pour plus d'informations, voir le *Modèle d'information* IEC 62541 décrit (voir 6.2.2.4).

La *Méthode* d'*interface de diagnostic* ne prend aucun paramètre d'entrée et retourne les paramètres d'entrée et de sortie du dernier appel de la *Méthode ReadSafetyData*.

De plus, un numéro de séquence à 2 octets est ajouté à l'*interface de diagnostic*, ce qui permet de détecter les appels manqués à la suite d'une interrogation. Le numéro de séquence compte le nombre d'accès aux *ReadSafetyData*.

Une meilleure pratique recommandée consiste à stocker tous les paramètres d'entrée et de sortie si SComErr_diag est <> 0.

7 Protocole de couche de communication de sécurité

7.1 Généralités

Le présent article décrit en détail le protocole utilisé pour la *couche de communication de sécurité*.

7.2 SafetyProvider et SafetyConsumer

7.2.1 Format des SPDU

7.2.1.1 Généralités

La Figure 11 représente la structure d'une *RequestSPDU* provenant du *SafetyConsumer*, et contient un *SafetyConsumerID*, un *MonitoringNumber* (*MNR*) et un octet de *Fanions* (*non relatifs à la sécurité*). Voir les 7.2.1.2 à 7.2.1.4 pour plus d'informations. Voir le 6.2.3.3 pour des détails sur la définition du *RequestSPDUDataType*.

NOTE 1 La *RequestSPDU* ne contient pas de signature *CRC*.

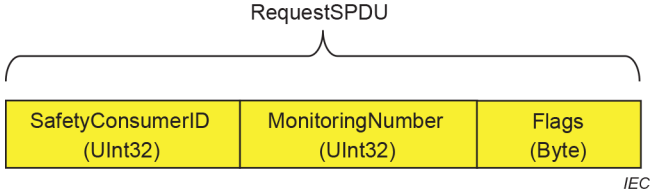


Figure 11 – RequestSPDU

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

La Figure 12 représente la structure d'une *ResponseSPDU* provenant du *SafetyProvider*, et contient les *SafetyData* (1 octet à 1 500 octets), un code de sécurité supplémentaire de 25 octets (*STrailer*) et les *NonSafetyData*. Voir les 7.2.1.5 à 7.2.1.11 pour plus d'informations. Voir le 6.2.3.4 pour des détails sur la définition du *ResponseSPDUDataType*.

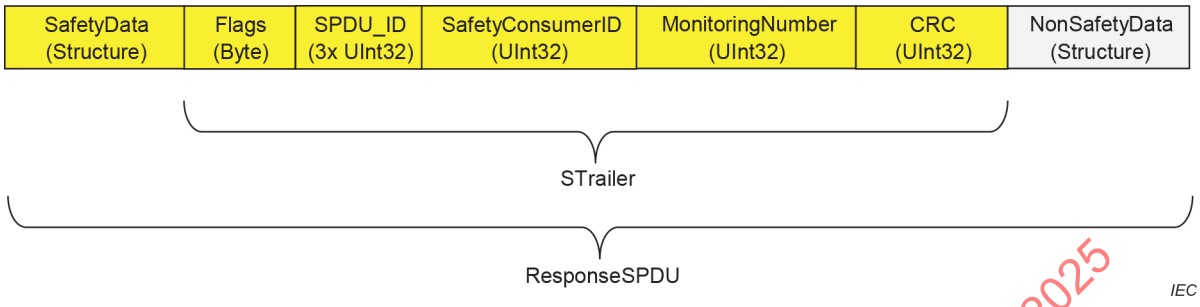


Figure 12 – ResponseSPDU

NOTE 2 Pour éviter les déclenchements parasites, la *ResponseSPDU* est transmise de manière atomique (cohérente) depuis l'*interface de plateforme OPC UA* du *SafetyProvider* vers l'*interface de plateforme OPC UA* du *SafetyConsumer*. Il s'agit de la tâche du *mappeur OPC UA* correspondant (voir Figure 2).

7.2.1.2 RequestSPDU: SafetyConsumerID

Identificateur de l'instance de *SafetyConsumer* à des fins de diagnostic (voir 9.1.2).

7.2.1.3 RequestSPDU: MonitoringNumber

Le *SafetyConsumer* utilise le *MNR* pour détecter des *SPDU* mal synchronisées, par exemple des *SPDU* qui sont continuellement répétées par un élément de réseau erroné qui stocke des données. Un *MNR* différent est utilisé dans chaque *RequestSPDU* d'un *SafetyConsumer* donné, et une *ResponseSPDU* n'est acceptée que si son *MNR* correspond au *MNR* de la *RequestSPDU* correspondante.

La vérification de l'exactitude du *MNR* est effectuée uniquement par le *SafetyConsumer*.

7.2.1.4 RequestSPDU: fanions

[RQ7.1] Les *fanions* du *SafetyConsumer* (*RequestSPDU.Flags*) doivent être utilisés comme cela est indiqué au 6.2.3.1.

7.2.1.5 ResponseSPDU: SafetyData

[RQ7.2] Les *SafetyData* doivent contenir les données d'application relatives à la sécurité transmises par le *SafetyProvider* au *SafetyConsumer*. Elles comprennent une ou plusieurs variables de base de l'IEC 62541 (voir 6.2.5). Afin de réduire les distinctions de cas, les *SafetyData* doivent toujours être une structure, même si elles ne comprennent qu'une seule *Variable* de base de l'IEC 62541.

Pour le calcul de la signature *CRC*, l'ordre dans lequel ces données sont traitées par le calcul est important. Le *SafetyProvider* et le *SafetyConsumer* doivent s'accorder sur le nombre, le type et l'ordre des données d'application transmises dans les *SafetyData*. La séquence des *SafetyData* est fixe.

Les *SafetyData* peuvent contenir des *qualificatifs* pour une activation fine des valeurs de substitution *Failsafe*. Pour des *valeurs de processus* valides, les *qualificatifs* correspondants sont définis sur 1 ("good"), tandis que la valeur 0 ("bad") est utilisée pour les valeurs non valides. Les *valeurs de processus* non valides sont remplacées par des *valeurs de substitution Failsafe* dans l'application de sécurité du *SafetyConsumer*. Voir le 6.3.6.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

7.2.1.6 ResponseSPDU: fanions

[RQ7.3] Les *fanions* du *SafetyProvider* (*ResponseSPDU.Flags*) doivent être utilisés comme cela est indiqué au 6.2.3.2.

[RQ7.4] Les *fanions* dans les *ResponseSPDU.Flags* réservés pour une utilisation future doivent être mis à zéro par le *SafetyProvider* et ne doivent pas être évalués par le *SafetyConsumer*.

7.2.1.7 ResponseSPDU: SPDU_ID

Ce champ est utilisé par le *SafetyConsumer* pour vérifier si la *ResponseSPDU* provient du bon *SafetyProvider*. Pour plus d'informations, voir le 7.2.3.1.

7.2.1.8 ResponseSPDU: SafetyConsumerID

[RQ7.5] Le *SafetyConsumerID* dans la *ResponseSPDU* doit être une copie du *SafetyConsumerID* reçu dans la *RequestSPDU* correspondante. Voir le 7.2.3.1.

7.2.1.9 ResponseSPDU: MonitoringNumber

[RQ7.6] Le *MonitoringNumber* dans la *ResponseSPDU* doit être une copie du *MonitoringNumber* reçu dans la *RequestSPDU* correspondante. Voir le 7.2.3.1.

NOTE Le *SafetyConsumer* utilise le *ResponseSPDU.MonitoringNumber* pour détecter les *SPDU* reçues avec une erreur d'opportunité, par exemple les *SPDU* qui sont continuellement répétées par un élément de réseau erroné qui stocke des données. Un *MonitoringNumber* différent est utilisé dans chaque *RequestSPDU* d'un *SafetyConsumer* donné, et une *ResponseSPDU* n'est acceptée que si son *MonitoringNumber* correspond au *MonitoringNumber* dans la *RequestSPDU* correspondante.

7.2.1.10 ResponseSPDU: CRC

[RQ7.7] Le *CRC* de la *ResponseSPDU* doit être utilisé pour détecter une corruption des données. Voir le 7.2.3.6 pour connaître son mode de calcul dans le *SafetyProvider* et son mode de vérification dans le *SafetyConsumer*.

7.2.1.11 ResponseSPDU: NonSafetyData

[RQ7.8] Cette structure doit être utilisée pour transmettre de manière cohérente des valeurs de *NonSafetyData* (par exemple, des informations de diagnostic) avec des données de sécurité. Les *NonSafetyData* ne sont pas protégées par un *CRC* et peuvent provenir d'une source dangereuse.

[RQ7.9] Lorsqu'elles sont présentées à l'application de sécurité (par exemple, à la sortie du *SafetyConsumer*), les valeurs non relatives à la sécurité doivent être clairement indiquées comme "non relatives à la sécurité" au moyen d'un mécanisme approprié spécifique au fournisseur (par exemple, à l'aide d'une couleur différente).

Pour éviter d'éventuels problèmes de structures vides, la structure factice *NonSafetyDataPlaceholder* doit être utilisée lorsqu'aucune *NonSafetyData* n'est utilisée (voir l'exigence RQ6.8).

7.2.2 Comportement

7.2.2.1 Généralités

Les deux services *SCL SafetyProvider* et *SafetyConsumer* sont spécifiés à l'aide de diagrammes d'états.

IECNORM.COM : Click to View the full PDF of IEC 62541-15:2025

7.2.2.2 Diagramme de séquence du SafetyProvider et du SafetyConsumer

La Figure 13 et la Figure 14 représentent respectivement les séquences de demandes et de réponses avec des *SafetyData* pour le présent document, en utilisant respectivement les mécanismes de communication *Client/Serveur* et *PubSub* IEC 62541. Les figures ne représentent que des scénarios choisis et sont donc informatives.

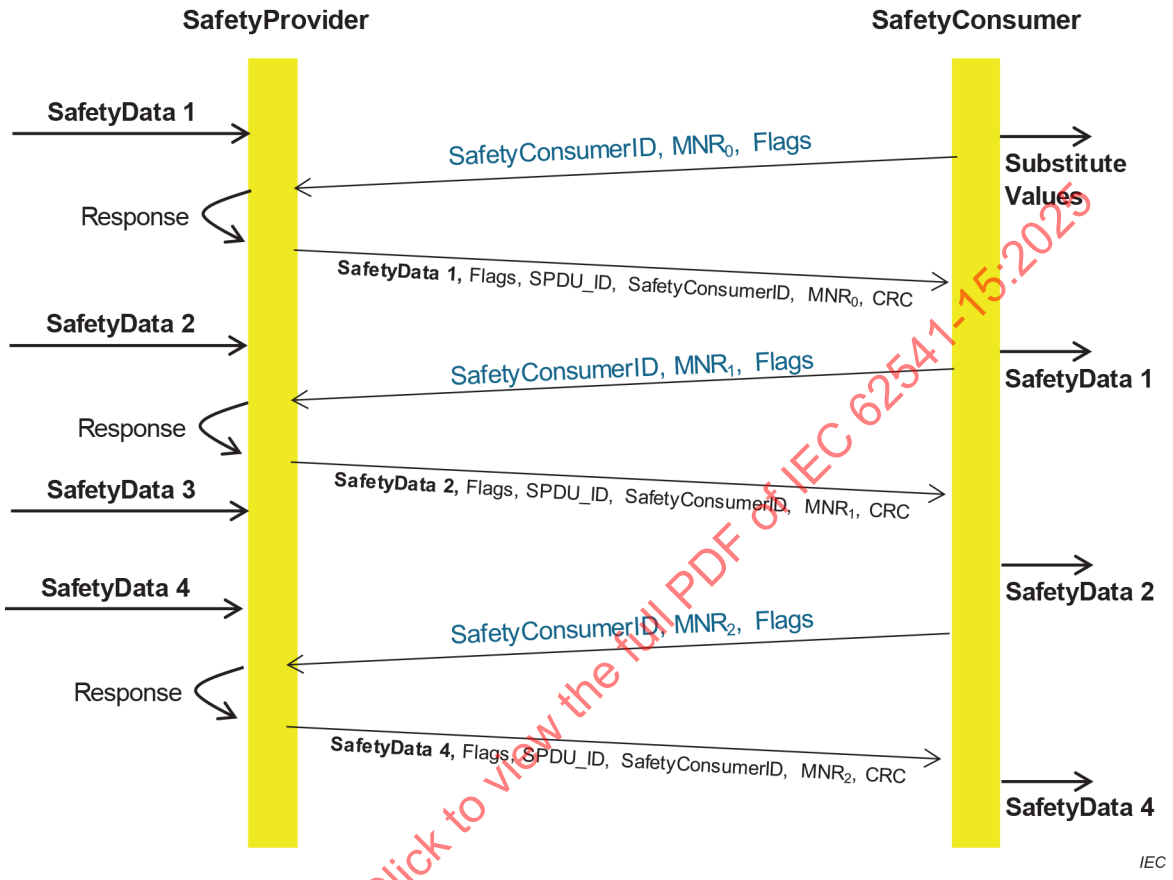


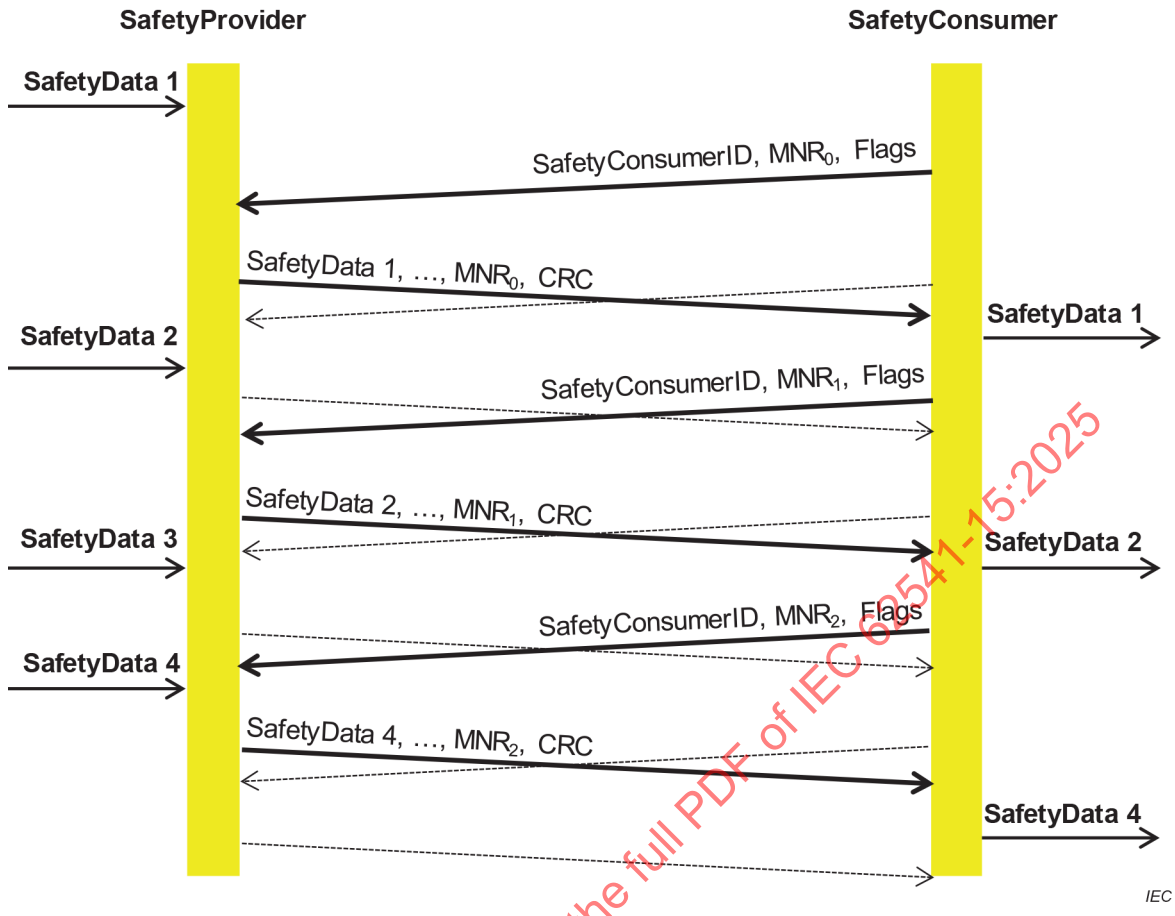
Figure 13 – Diagramme de séquence des demandes et réponses (Client/Serveur)

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

[RQ7.10] Dans le cas d'une communication *Client/Serveur*, le *mappeur OPC UA* d'un *SafetyConsumer* peut appeler plusieurs fois de suite un *SafetyProvider* (soit la mise en œuvre du diagramme d'états elle-même, soit le *mappeur OPC UA* du *SafetyProvider*) avec une *RequestSPDU* identique. Dans ce cas, le *SafetyProvider* (diagramme d'états ou *mappeur OPC UA*) doit répondre à toutes les demandes. Les *ResponseSPDU* retournées peuvent contenir différentes valeurs (par exemple, les *valeurs de processus* actuellement disponibles) ou contenir les valeurs initialement retournées.

[RQ7.11] Pour chaque *SafetyProvider*, le choix de comportement mis en œuvre conformément à la RQ7.10 (c'est-à-dire si les *valeurs de processus* actuellement disponibles ou les valeurs initialement retournées sont utilisées) doit être documenté dans le manuel de sécurité correspondant.

NOTE 1 Étant donné qu'un *SafetyConsumer* vérifie la présence de *MNR* modifiés dans les *ResponseSPDU* (voir l'état S14_WaitForChangedSPDU du *SafetyConsumer* dans le Tableau 34 et la Transition T17 dans le Tableau 35), il n'utilise que la première *ResponseSPDU* évaluée pour un *MNR* donné, et toutes les autres *ResponseSPDU* avec le même *MNR* sont ignorées.



NOTE Les flèches en gras représentent une communication avec de nouvelles valeurs de données, tandis que les flèches en pointillés contiennent des valeurs de données répétées.

Figure 14 – Diagramme de séquence des demandes et réponses (PubSub)

NOTE 2 Les diagrammes d'états conformes au présent document ne comportent aucun mécanisme de nouvelle tentative permettant d'augmenter la tolérance aux anomalies. En revanche, il est admis par hypothèse que la nouvelle tentative est déjà traitée dans la pile de l'IEC 62541 (par exemple, utilisation de la communication *Client/Serveur* ou choix d'un taux de mise à jour plus élevé pour *PubSub*). Les lignes en pointillés ne font donc pas partie du présent document, mais symbolisent l'envoi répété de données mis en œuvre dans la pile de l'IEC 62541.

Le *SafetyConsumerTimeout* est le temps de fonctionnement du chien de garde vérifié dans le *SafetyConsumer*. Le chien de garde est redémarré immédiatement avant la génération d'une nouvelle *RequestSPDU* (Transitions T14 et T28 du *SafetyConsumer* dans le Tableau 35). Si une *ResponseSPDU* appropriée est reçue dans les temps et que les contrôles d'intégrité, d'authenticité et d'opportunité des données sont tous valides, le temporisateur n'expire pas avant le redémarrage.

Sinon, le temporisateur du chien de garde expire et le *SafetyConsumer* déclenche une réaction de sécurité. Pour une vérification correcte de son temporisateur, le *SafetyConsumer* est exécuté de manière cyclique, avec une période *ConsumerCycleTime*. Le *ConsumerCycleTime* est présumé inférieur à la valeur *SafetyConsumerTimeout*.

Le *ConsumerCycleTime* est le temps maximal de mise à jour cyclique du *SafetyConsumer*. Il s'agit du délai entre une exécution du *SafetyConsumer* et la prochaine exécution du *SafetyConsumer*. La mise en œuvre de la surveillance du *ConsumerCycleTime* et la réaction en cas de dépassement du *ConsumerCycleTime* ne font pas partie du présent document; elles sont spécifiques au fournisseur.

[RQ7.12] Le *ConsumerCycleTime* doit être surveillé selon une approche relative à la sécurité.

[RQ7.26] Une modification de la valeur du paramètre *SafetyConsumerTimeout* doit entraîner la modification immédiate du *ConsumerTimer*.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

7.2.2.3 Durée de la demande

S'il est nécessaire de s'assurer que des *SafetyData* données (par exemple, une sollicitation de *sécurité* ou une valeur de commande) provenant de l'application de sécurité du *SafetyProvider* sont reçues par un *SafetyConsumer* et transférées à l'application de *sécurité* du *SafetyConsumer*, c'est-à-dire si aucune des *SafetyData* d'une série ne doit être ignorée, les deux cas suivants doivent être examinés.

Dans le cas A, le *SafetyProvider* répond à des *RequestSPDU* identiques répétées avec des *ResponseSPDU* qui contiennent les valeurs initialement retournées. Ce cas s'adresse à la communication *PubSub* et constitue une option pour la communication *Client/Serveur* (voir RQ7.11). Dans ce cas, l'application de sécurité du *SafetyProvider*' doit fournir les *SafetyData* correspondantes à la *SAPI* du *SafetyProvider* jusqu'à ce qu'au moins une modification du *MNR* soit détectée.

Dans le cas B, le *SafetyProvider* répond à des *RequestSPDU* identiques répétées avec les *SafetyData* actuellement disponibles. Il s'agit d'une option pour la communication *Client/Serveur* (voir RQ7.11). Dans ce cas, l'application de *sécurité* du *SafetyProvider* doit fournir les *SafetyData* correspondantes à la *SAPI* du *SafetyProvider* jusqu'à ce qu'au moins deux modifications du *MNR* soient détectées.

La Figure 15 et la Figure 16 donnent des exemples qui justifient le cas B en décrivant deux séquences de *ResponseSPDU* envoyées par un *SafetyProvider*. Du fait de la non-synchronisation des cycles du *SafetyProvider* et du *SafetyConsumer*, un *SafetyConsumer* peut évaluer une *ResponseSPDU* quelconque parmi un nombre donné pour une *RequestSPDU* donnée.

Dans les exemples, les *SafetyData* sont constituées de deux composantes: les "données de *sécurité* correspondantes", c'est-à-dire une sollicitation de *sécurité* qui ne doit pas être ignorée (une des valeurs "A", "B" ou "C") et des valeurs de mesure numériques non sollicitées dont il importe peu que certaines ne soient pas reçues par le *SafetyConsumer*.

Étant donné que les *NonSafetyData* sont transmises de manière cohérent avec les *SafetyData*, les mêmes considérations s'appliquent pour les *NonSafetyData*.

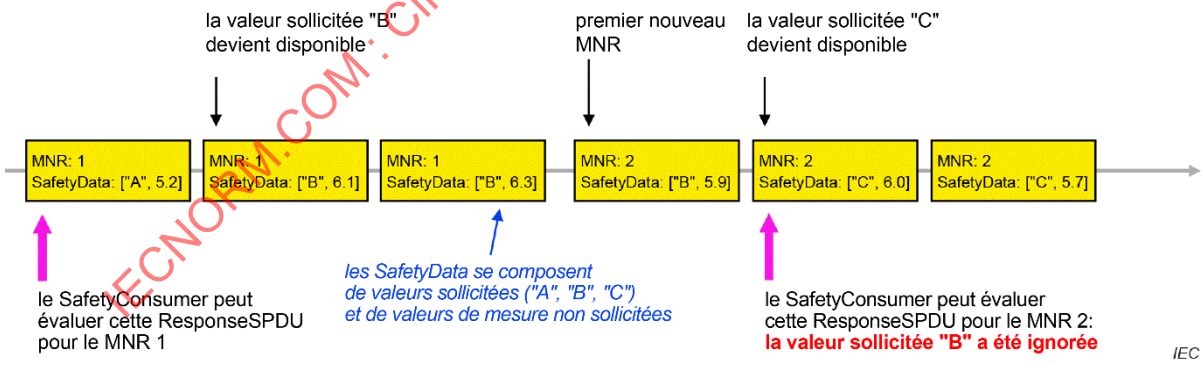


Figure 15 – Exemple de durée de sollicitation pour une valeur sollicitée ignorée lorsque les SafetyData actuellement disponibles ne sont pas fournies tant qu'une deuxième modification du MNR n'a pas lieu

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

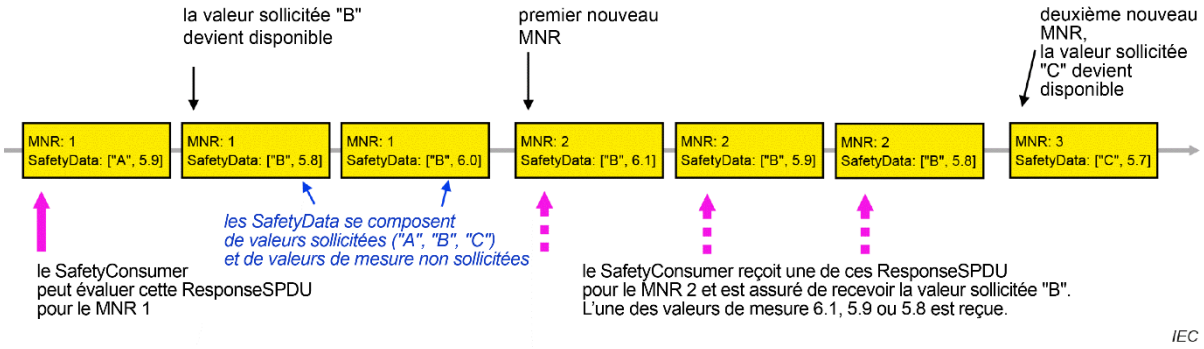


Figure 16 – Exemple de durée de sollicitation pour une valeur sollicitée reçue lorsque les SafetyData actuellement disponibles sont fournies

7.2.2.4 Diagramme d'états du SafetyProvider

[RQ7.13] La Figure 17 donne une représentation simplifiée du diagramme d'états du *SafetyProvider*. Le comportement exact est décrit dans le Tableau 30, le Tableau 31 et le Tableau 32. Le *SafetyProvider* doit mettre en œuvre ce comportement. Il n'est pas tenu de suivre littéralement les entrées données dans les tableaux si le comportement observable extérieurement ne varie pas.

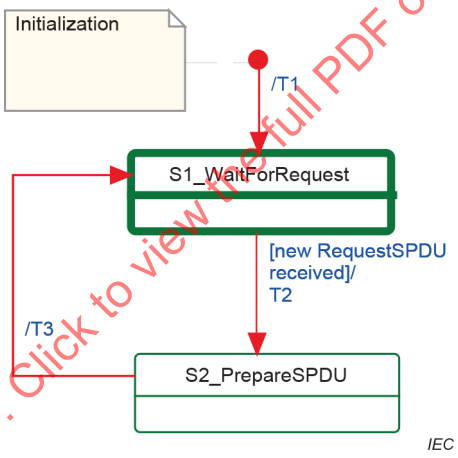
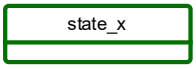
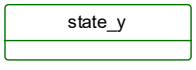


Figure 17 – Représentation simplifiée du diagramme d'états du SafetyProvider

Le Tableau 29 fournit les symboles utilisés pour les diagrammes d'états.

Tableau 29 – Symboles utilisés pour les diagrammes d'états

Représentation graphique	Type	Description
	État d'activité	Dans ces états d'activité qu'il est possible d'interrompre, le *SafetyProvider* attend une nouvelle entrée.
	État d'action	Dans ces états d'action qu'il n'est pas possible d'interrompre, les événements comme les nouvelles demandes sont différés jusqu'à ce que l'état d'activité suivant soit atteint (voir [1][1]).

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

1 Les chiffres entre crochets renvoient à la Bibliographie.

Les transitions sont déclenchées en cas d'événement. Le terme "événement" dans ce contexte signifie soit un appel de *Méthode* dans le cas d'une communication *Client/Serveur*, soit la détection d'une *RequestSPDU* modifiée par le *mappeur OPC UA* dans le cas d'une communication *PubSub*.

Si plusieurs transitions sont possibles, les conditions de garde (voir [...] dans les diagrammes UML) définissent la transition à déclencher.

Le diagramme est composé d'états d'activité et d'action. Les états d'activité sont encadrés par des lignes en gras, et les états d'action sont encadrés par des lignes minces. Les états d'activité peuvent être interrompus par de nouveaux événements, ce qui n'est pas le cas des états d'action. Les événements externes qui se produisent lorsque le diagramme d'états est dans un état d'action sont différés jusqu'à ce que l'état d'activité suivant soit atteint.

NOTE Les détails sur la manière de mettre en œuvre les états d'activité et les états d'action sont spécifiques au fournisseur. Généralement, dans un système en temps réel, la tâche de réalisation du diagramme d'états du *SafetyProvider* ou du *SafetyConsumer* est exécutée de manière cyclique (voir 6.3.5). Lorsque la tâche est réactivée par le planificateur du système d'exploitation lorsqu'il est dans un état d'action, il exécute les états d'action jusqu'à ce que son intervalle de temps soit épuisé ou qu'un état d'activité soit atteint. Lorsqu'une tâche se trouvant dans un état d'activité est réactivée, elle vérifie l'entrée. Si aucune nouvelle entrée n'est disponible, elle retourne immédiatement à l'état de veille sans changement d'état.

Si une entrée est disponible, elle lance l'exécution des états d'action jusqu'à ce sont intervalle de temps soit disponible ou jusqu'à ce que l'état d'activité suivant soit atteint.

Le Tableau 30 décrit les éléments internes d'une instance de *SafetyProvider*.

Tableau 30 – Éléments internes d'une instance de SafetyProvider

Éléments internes	Type	Définition
RequestSPDU_i	Variable	Mémoire locale pour la *RequestSPDU* (exigée pour réagir aux modifications).
SPDU_ID_1_i	UInt32	Variable locale pour stocker *SPDU_ID_1*.
SPDU_ID_2_i	UInt32	Variable locale pour stocker *SPDU_ID_2*.
SPDU_ID_3_i	UInt32	Variable locale pour stocker *SPDU_ID_3*.
BaseID_i	Guid	Variable locale contenant le BaseID (tiré de la *SPI* ou de la *SAPI*).
ProviderID_i	UInt32	Variable locale contenant le ProviderID (tiré de la *SPI* ou de la *SAPI*).
<Get RequestSPDU>	Macro	Instruction de prendre l'ensemble de la *RequestSPDU* du *mappeur OPC UA*.
<Set ResponseSPDU>	Macro	Instruction de transférer l'ensemble de la *ResponseSPDU* au *mappeur OPC UA*.

<Calc SPDU_ID_i>	Macro	const uint32 SafetyProviderLevel_ID:= ... // voir le 7.2.3.3
		
		If(SAPI.SafetyBaseID == 0) then
		BaseID_i:= SPI.SafetyBaseIDConfigured
		Else
		BaseID_i:= SAPI.SafetyBaseID
		
		Endif
		If(SAPI.SafetyProviderID == 0) then
		ProviderID_i:= SPI.SafetyProviderIDConfigured
		Else
		ProviderID_i:= SAPI.SafetyProviderID
		
		Endif
		
		SPDU_ID_1_i:= BaseID_i (octets 0...3)
		XOR SafetyProviderLevel_ID
		
		SPDU_ID_2_i:= BaseID_i (octets 4...7)
		XOR SPI.SafetyStructureSignature
		
		SPDU_ID_3_i:= BaseID_i (octets 8...11)
		XOR BaseID_i (octets 12...15)
		XOR ProviderID_i
		
		// voir le 7.2.3.2 pour une clarification

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025

Éléments internes	Type	Définition
<build ResponseSPDU>	Macro	Prendre le *MNR* et le *SafetyConsumerID* de la *RequestSPDU* reçue. Ajouter les attributs SPDU_ID_1_i, SPDU_ID_2_i et SPDU_ID_3_i, les fanions, les *SafetyData* et les *NonSafetyData*, ainsi que le *CRC* calculé. Voir le 7.2.3.1

Le Tableau 31 décrit les états d'une instance de *SafetyProvider*. Le *SafetyProvider* ne vérifie pas si la configuration est correcte. Il répond aux demandes même s'il est mal configuré (par exemple, si son *SafetyProviderID* a une valeur de zéro). Cependant, les *SafetyConsumers* ne tentent jamais de communiquer avec les *SafetyProviders* ayant des paramètres incorrects (voir les Transitions T13 et T27 dans le Tableau 35 et la macro <ParametersOK?> dans le Tableau 33).

Tableau 31 – États d'une instance de SafetyProvider

Nom d'état	Description d'état
Initialization	// État initial SAPI.SafetyData:= 0 SAPI.NonSafetyData:= 0 SAPI.MonitoringNumber:= 0 SAPI.SafetyConsumerID:= 0 SAPI.OperatorAckRequested:= 0 RequestSPDU_i:= 0
S1_WaitForRequest	// attente de la prochaine *RequestSPDU* du *SafetyConsumer* <Get RequestSPDU>
S2_PrepareSPDU	ResponseSPDU.Flags.ActivateFSV:= SAPI.ActivateFSV ResponseSPDU.Flags.OperatorAckProvider:= SAPI.OperatorAckProvider ResponseSPDU.Flags.TestModeActivated:= SAPI.EnableTestMode <Calc SPDU_ID_i> <build ResponseSPDU> // voir le 7.2.3.1

Le Tableau 32 décrit les transitions du *SafetyProvider*.

Tableau 32 – Transitions du SafetyProvider

Transition	État source	État cible	Condition de garde	Activité
T1	Init	S1		
T2	S1	S2	**// Réception de la RequestSPDU**[1] -	// **Traitement de la demande** RequestSPDU_i:= RequestSPDU SAPI.MonitoringNumber:= RequestSPDU.MonitoringNumber SAPI.SafetyConsumerID:= RequestSPDU.SafetyConsumerID SAPI.OperatorAckRequested:= RequestSPDU.Flags.OperatorAckRequested
T3	S2	S1	// **Préparation de la SPDU** -	<Set ResponseSPDU>

[1] Voir l'explication précédente au 7.2.2.3 des événements qui déclenchent cette transition.

IECNORM.COM : Click to view the full PDF of IEC 62541-15:2025