

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Field device integration (FDI) –
Part 7: Communication devices**

**Intégration des appareils de terrain (FDI) –
Partie 7: Appareils de communication**



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2021 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 Central Office
3, rue de Varembe
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 It also gives information on projects, replaced and withdrawn publications.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 online collection - oc.iec.ch

Discover our powerful search engine and read freely all the publications previews. 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 22 000 terminological entries in English and French, with equivalent terms in 18 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 online collection - oc.iec.ch

Découvrez notre puissant moteur de recherche et consultez gratuitement tous les aperçus des publications. 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 22 000 articles terminologiques en anglais et en français, ainsi que les termes équivalents dans 16 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**Field device integration (FDI) –
Part 7: Communication devices**

**Intégration des appareils de terrain (FDI) –
Partie 7: Appareils de communication**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

ICS 25.040.40; 35.100.05

ISBN 978-2-8322-9314-0

**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.**

CONTENTS

FOREWORD	6
INTRODUCTION	8
1 Scope	9
2 Normative references	10
3 Terms, definitions, abbreviated terms and conventions	10
3.1 Terms and definitions	10
3.2 Abbreviated terms	10
3.3 Conventions	11
4 General	11
5 FDI Communication Package	13
5.1 General	13
5.2 EDD	13
5.2.1 General rules	13
5.2.2 Device component	14
5.2.3 CommunicationDevice component	15
5.2.4 Communication service provider component	16
5.2.5 Connection Point component	17
5.2.6 Connection Point collection	18
5.2.7 Network component	18
5.2.8 ValidateNetwork	20
5.2.9 ValidateModules	20
5.2.10 UIP specifics	21
5.2.11 Deployment	21
6 Communication relations	21
7 FDI Communication Server definition	23
7.1 General	23
7.2 General characteristics	23
7.3 Information Model	23
7.3.1 General	23
7.3.2 CommunicationServerType	26
7.3.3 ServerCommunicationDeviceType	30
7.3.4 ServerCommunicationServiceType	33
7.4 OPC UA Server Profile for FDI Communication Server	37
7.5 Mapping the FDI Server IM to the FDI Communication Server IM	38
7.5.1 General	38
7.5.2 Information Model differences	38
7.6 Installer	39
7.7 FDI Communication Package	39
7.7.1 General	39
7.7.2 EDD for Lightweight Communication Server	40
7.7.3 EDD for Multi-Channel Communication Server	40
7.7.4 COMMANDs in EDDs for FDI Communication Servers	41
7.7.5 Documentation	41
7.8 Handling and behavior	41
7.8.1 General	41

7.8.2	Deployment	42
7.8.3	Server configuration	42
7.8.4	Start up	43
7.8.5	Shutdown	43
7.8.6	Watchdog	43
7.8.7	Establish the OPC UA connection	43
7.8.8	Instantiate the Communication Server	44
7.8.9	Configure the communication hardware	44
7.8.10	Configuring the Network	44
7.8.11	Parameterize	44
7.8.12	Initialize	44
7.8.13	Create the communication service object	45
7.8.14	Communication relation	45
7.8.15	Connect	45
7.8.16	Disconnect	46
7.8.17	Abort Indication	46
7.8.18	Scan	46
7.8.19	SetAddress	46
8	FDI Communication Gateway definition	46
8.1	General	46
8.2	Information Model	46
8.2.1	General	46
8.2.2	CommunicationGatewayType	47
8.2.3	GatewayCommunicationDeviceType	48
8.2.4	GatewayCommunicationServiceType	51
8.3	FDI Communication Package	55
8.3.1	General	55
8.3.2	EDD	56
8.4	Handling and behavior	57
8.4.1	General	57
8.4.2	Deployment	58
8.4.3	Start up	58
8.4.4	Configure the communication hardware	58
8.4.5	Configure the Network	58
8.4.6	Parameterize	58
8.4.7	Communication relation	59
8.4.8	Connect	59
8.4.9	Disconnect	59
8.4.10	Abort indication	59
8.4.11	Scan	59
8.4.12	Communication Error Handling	60
Annex A (informative)	Layered protocols	61
A.1	General	61
A.2	Convention for protocol-specific annex creation	61
A.3	FDI Communication Package definition	62
A.3.1	Communication services	62
A.3.2	Connection Point	62
A.3.3	Network	62
A.4	Representation in the IM	63

Annex B (normative) Namespace and mappings	64
Figure 1 – FDI architecture diagram	9
Figure 2 – FDI communication infrastructure architecture	12
Figure 3 – Communication relation	22
Figure 4 – Communication relation state chart	23
Figure 5 – FDI Communication Server AddressSpace	25
Figure 6 – CommunicationServerType	26
Figure 7 – ServerCommunicationDeviceType	30
Figure 8 – ServerCommunicationServiceType	34
Figure 9 – Information Model differences (example)	38
Figure 10 – FDI Communication Server state machine	42
Figure 11 – Communication relation state chart	45
Figure 12 – Gateway information model	47
Figure 13 – CommunicationGatewayType	48
Figure 14 – GatewayCommunicationDeviceType	49
Figure 15 – GatewayCommunicationServiceType	52
Figure 16 – Nested Communication	58
Table 1 – ValidateNetwork Action arguments	20
Table 2 – ValidateModules Action arguments	21
Table 3 – CommunicationServerType definition	26
Table 4 – MethodSet of CommunicationServerType	26
Table 5 – Reset Method arguments	27
Table 6 – Reset Method AddressSpace definition	27
Table 7 – Initialize Method arguments	28
Table 8 – Initialize Method AddressSpace definition	28
Table 9 – AddComponent Method arguments	29
Table 10 – AddComponent Method AddressSpace definition	29
Table 11 – RemoveComponent Method arguments	30
Table 12 – RemoveComponent Method AddressSpace definition	30
Table 13 – ServerCommunicationDeviceType definition	31
Table 14 – MethodSet of ServerCommunicationDeviceType	31
Table 15 – Scan Method arguments	32
Table 16 – Scan Method AddressSpace definition	32
Table 17 – ResetScan Method arguments	32
Table 18 – ResetScan Method AddressSpace definition	33
Table 19 – SetAddress Method arguments	33
Table 20 – ServerCommunicationServiceType definition	34
Table 21 – MethodSet of ServerCommunicationServiceType	34
Table 22 – Connect Method arguments	35
Table 23 – Disconnect Method arguments	36
Table 24 – Transfer Method arguments	36

Table 25 – GetPublishedData Method arguments.....	37
Table 26 – FDICommunicationServer_Facet definition	37
Table 27 – CommunicationGatewayType definition	48
Table 28 – GatewayCommunicationDeviceType definition.....	49
Table 29 – MethodSet of GatewayCommunicationDeviceType	49
Table 30 – Scan Method arguments.....	50
Table 31 – Scan Method AddressSpace definition.....	50
Table 32 – ScanNext Method arguments.....	51
Table 33 – ScanNext Method AddressSpace definition	51
Table 34 – GatewayCommunicationServiceType definition.....	52
Table 35 – MethodSet of GatewayCommunicationServiceType	53
Table 36 – Connect Method arguments.....	54
Table 37 – Transfer Method arguments.....	55

INTERNATIONAL ELECTROTECHNICAL COMMISSION

FIELD DEVICE INTEGRATION (FDI) –

Part 7: Communication devices

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

International Standard IEC 62769-7 has been prepared by subcommittee 65E: Devices and integration in enterprise systems, of IEC technical committee 65: Industrial-process measurement, control and automation.

This second edition cancels and replaces the first edition published in 2015. This edition constitutes a technical revision.

This edition includes the following significant technical changes with respect to the previous edition:

- a) support for generic protocol extension for faster adoption of other technologies;
- b) support of new protocols;
- c) generic protocol extension to allow adoption of other communication protocols;
- d) based on generic protocol extension: Modbus RTU.

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

FDIS	Report on voting
65E/764/FDIS	65E/774/RVD

Full information on the voting for the approval of this International Standard can be found in the report on voting indicated in the above table.

This document has been drafted in accordance with the ISO/IEC Directives, Part 2.

A list of all parts in the 62769 series, published under the general title *Field Device Integration (FDI)*, 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 "<http://webstore.iec.ch>" in the data related to the specific document. At this date, the document will be

- reconfirmed,
- withdrawn,
- replaced by a revised edition, or
- amended.

IMPORTANT – The 'colour inside' logo on the cover page of this publication indicates that it contains colours which are considered to be useful for the correct understanding of its contents. Users should therefore print this document using a colour printer.

INTRODUCTION

The IEC 62769 series has the general title *Field Device Integration (FDI)* and the following parts:

- Part 1: Overview
- Part 2: FDI Client
- Part 3: FDI Server
- Part 4: FDI Packages
- Part 5: FDI Information Model
- Part 6: FDI Technology Mapping
- Part 7: FDI Communication Devices
- Part 100: Profiles – Generic Protocol Extensions
- Part 101-1: Profiles – Foundation Fieldbus H1
- Part 101-2: Profiles – Foundation Fieldbus HSE
- Part 103-1: Profiles – PROFIBUS
- Part 103-4: Profiles – PROFINET
- Part 109-1: Profiles – HART and WirelessHART
- Part 115-2: Profiles – Protocol-specific Definitions for Modbus RTU
- Part 150-1: Profiles – ISA 100.11a

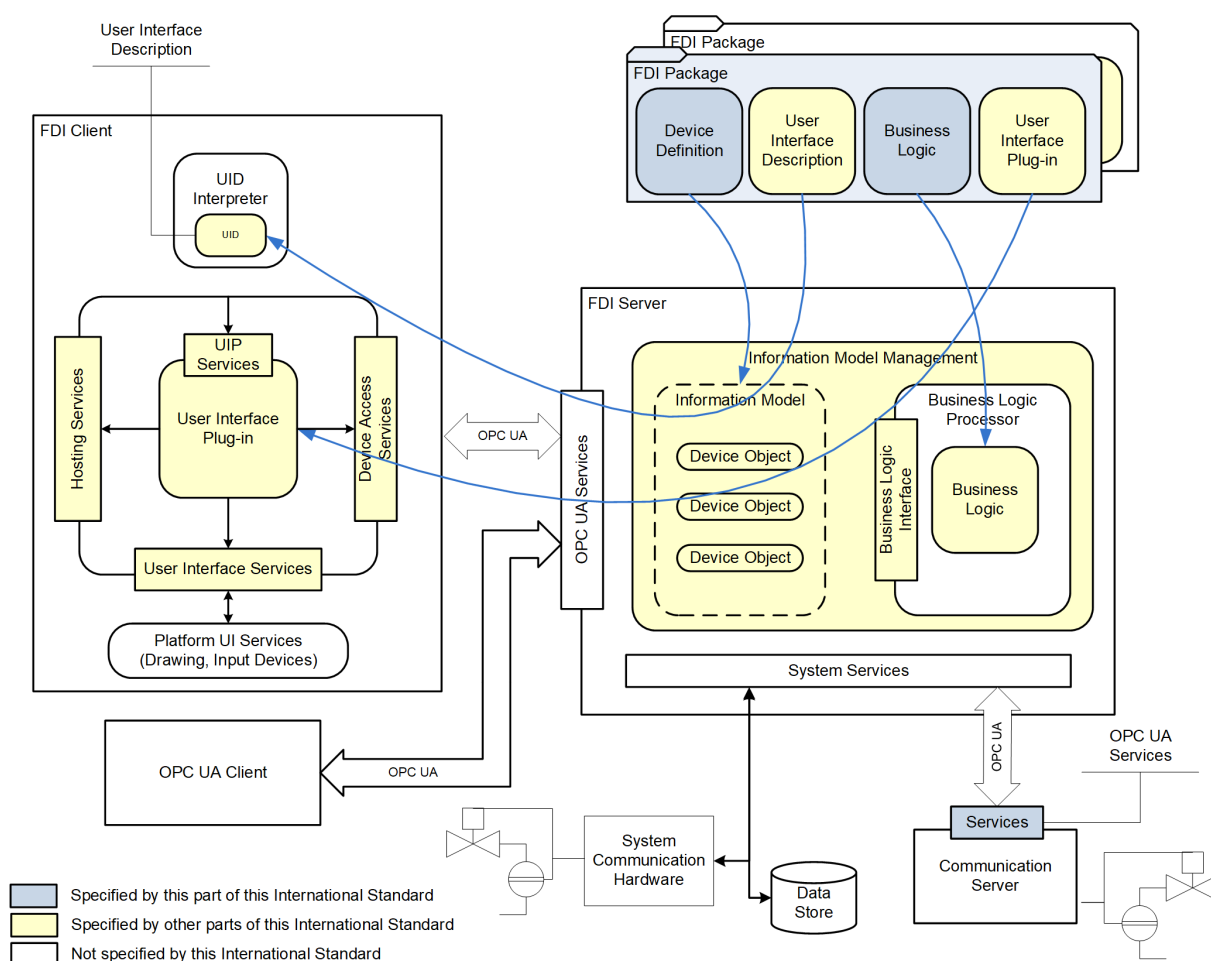
FIELD DEVICE INTEGRATION (FDI) –

Part 7: Communication devices

1 Scope

This part of IEC 62769 specifies the elements implementing communication capabilities called Communication Devices (IEC 62769-5).

The overall FDI architecture is illustrated in Figure 1. The architectural components that are within the scope of this document have been highlighted in this illustration. The document scope with respect to FDI Packages is limited to Communication Devices. The Communication Server shown in Figure 1 is an example of a specific Communication Device.



IEC

Figure 1 – FDI architecture diagram

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 61804-3, *Function blocks (FB) for process control and Electronic Device Description Language (EDDL) – Part 3: EDDL syntax and semantics*

IEC 61804-4, *Function blocks (FB) for process control and Electronic Device Description Language (EDDL) – Part 4: EDD interpretation*

IEC 62541 (all parts), *OPC Unified Architecture*

IEC 62769-1, *Field Device Integration (FDI) – Part 1: Overview*

IEC 62769-2, *Field Device Integration (FDI) – Part 2: FDI Client*

IEC 62769-3, *Field Device Integration (FDI) – Part 3: FDI Server*

IEC 62769-4:2020, *Field Device Integration (FDI) – Part 4: FDI Packages*

IEC 62769-5, *Field Device Integration (FDI) – Part 5: FDI Information Model*

3 Terms, definitions, abbreviated terms and conventions

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC 62769-1 as well as the following apply.

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

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

3.1.1

gateway

communication device that enables to bridge between different physical networks or different protocols

3.2 Abbreviated terms

For the purposes of this document, the abbreviated terms given in IEC 62769-1 and the following apply.

HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
PHY	Physical communication hardware
SNMP	Simple Network Management Protocol
TCP	Transmission Control Protocol
URI	Uniform Resource Identifier

3.3 Conventions

For the purposes of this document, the conventions given in IEC 62769-1 apply.

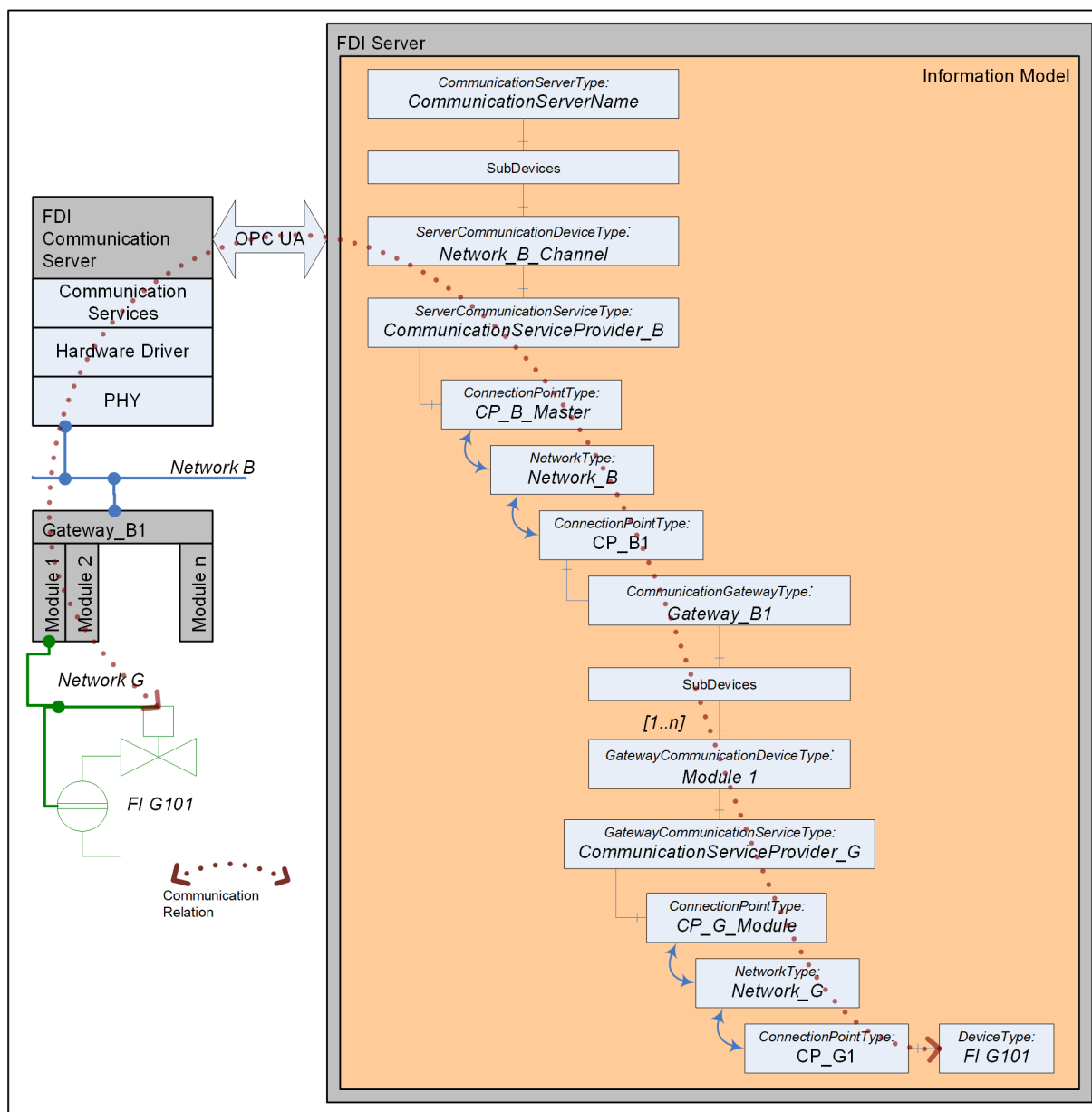
This document uses the graphical notation defined in IEC 62769-5.

4 General

The abstract term "FDI Communication Device" represents an entity implementing communication functions over a network using a specific protocol. The group of FDI Communication Devices splits into two main groups.

- a) The FDI Communication Server is a dedicated OPC UA Server providing access to one or more field device networks. The FDI Communication Server is specified in Clause 7.
- b) The FDI Communication Gateway enables to bridge between different physical networks or different protocols. The bridging business logic is implemented in the EDD component that is provided with an FDI Communication Package. The FDI Communication Gateway is specified in Clause 8.

NOTE The main differences between a Gateway and a Communication Server are: in terms of FDI, the FDI Communication Server is a dedicated OPC UA Server providing access to one or more field device networks. A Gateway is a communication device that enables to bridge between different physical networks or different protocols. The logical representation of a Gateway device within the FDI Server hosted Information Model enables the FDI Server to process communication in heterogeneous network topologies.



IEC

Figure 2 – FDI communication infrastructure architecture

The FDI Server hosted Information Model contains a representation of the network topology. (see also IEC 62769-5). The Information Model shown in Figure 2 is an example excerpt to illustrate how the used elements of the Information Model reflect the actual network topology.

- 1) The instance of *CommunicationServerType* (named *CommunicationServerName*) represents the FDI Communication Server. The FDI Communication Server implements physical communication network access (Communication hardware). Clause 7 describes related Information Model specifics, required FDI Communication Package content and handling of elements therein. For sub-devices, see IEC 62769-5.
- 2) The instance of *ServerCommunicationDeviceType* and *ServerCommunication-ServiceType* (named *Network_B_Channel*) maps to the FDI Communication Server implemented communication services. The *ServerCommunicationDeviceType* is specified in 7.3.3. The *ServerCommunicationServiceType* is specified in 7.3.4.

- 3) The instance of `CommunicationGatewayType` (named `Gateway_B1`) represents the physical Gateway. Clause 8 describes the related Information Model specifics, the required FDI Package content and the handling of elements therein.
- 4) The instance of `GatewayCommunicationDeviceType` (named `Module 1`) maps to a physical or logical module enabling communication to the network to which this module is connected. The `GatewayCommunicationDeviceType` is specified in 8.3.2.3. The related Gateway specifics are described in Clause 8.
- 5) The instance of `GatewayCommunicationServiceType` (named `CommunicationServiceProvider_G`) represents the Gateways' ability to process communication services. The Gateway-specific implementation of `GatewayCommunicationServiceType` is based on Business Logic that enables to run communication services in heterogeneous communication networks.
- 6) A communication relation (more details are described in Clause 6) between a physical device and the device representation managed by the FDI Server is always associated to communication service objects that are instances of a `GatewayCommunicationServiceType` or `ServerCommunicationServiceType`. The ability of instantiating multiple communication service objects supports protocols enables to operate multiple logical connections between a bus master and a device.
- 7) The Information Model represents the connections between the physical devices shown on the left side of Figure 2 based on instances of `ConnectionPointType` `NetworkType` and the depicted relations. `ConnectionPointType` and `NetworkType` are specified in IEC 62769-5.

5 FDI Communication Package

5.1 General

The FDI Server imports the FDI Communication Package like any other FDI Device Package. Clause 5 specifies the FDI Communication Package details.

5.2 EDD

5.2.1 General rules

The FDI Communication Package contained EDD is not restricted, but bound to a protocol-specific annex (IEC 62769-4:2020, Annex F).

The EDD elements as specified in IEC 62769-4:2020, Annex F, and provided with an FDI Communication Package shall describe:

- a) Parameter and parameter structures. Mandatory protocol-specific parameter definitions are found in IEC 62769-4:2020, Annex F. The parameters shall contain any parameter that requires adjustment for proper communication service operation.
- b) Physical Layer identification. Protocol-specific definitions are found in IEC 62769-4:2020, Annex F.

Communication devices' modularity: The modularity information shall be based on using the EDDL constructs `COMPONENT` (see IEC 61804-3).

FDI envisions communication device modularity to cope with communication hardware providing multiple physical or logical communication channels to access multiple logical or physical communication networks. Each module element of the whole communication device shall be described by a separate EDD element.

- c) The `COMPONENT` definition shall be used to support the system implemented topology configuration. Protocol-specific definitions are found in IEC 62769-4:2020, Annex F. The related `COMPONENT` definitions are described in 5.2.2, 5.2.3, 5.2.4, and 5.2.7.
- d) The Business Logic shall contain a method enabled to validate the network (see 5.2.8). The validation function considers the elements only directly connected to the network. The validation function shall be referred by the EDDL specified `CHECK_CONFIGURATION` attribute.

- e) The Business Logic can contain a method enabled to validate the module configuration (see 5.2.9) or the network configuration (see 5.2.8). The validation function considers the elements only directly connected to the related parent element in the topology. The validation function shall be referred by the EDDL specified CHECK_CONFIGURATION attribute.
- f) Connection Point data: The Connection Point (see 5.2.4 and 5.2.6) shall be described through EDDL constructs COMPONENT, COLLECTION and VARIABLE. The COMPONENT definition associates the Connection Point element to the Communication Device. The VARIABLE definitions represent the properties of a specific Connection Point. The COLLECTION represents the Connection Point structure as such. Protocol-specific definitions are found in IEC 62769-4:2020 Annex F.
- g) MENU:
The Menu structure shall follow the Menu conventions for PC-based applications in accordance with IEC 61804-4, which enables access to:
 - 1) FDI Communication Device Type (Bus) parameters: These parameters shall be made accessible by means of "offline_root_menu".
 - 2) Topology Configuration Dialogs shall be made available by means of the menu entry point "topology_configuration".

5.2.2 Device component

Each FDI Communication Package shall contain an EDD element describing the device.

```
COMPONENT <DeviceComponentId>
{
    LABEL "<Label>";
    CAN_DELETE TRUE;
    CHECK_CONFIGURATION <ValidateModules>;
    CLASSIFICATION NETWORK_COMPONENT;
    COMPONENT_RELATIONS
    {
        <CommunicationDeviceRelationId>
    }
}

COMPONENT_RELATION <CommunicationDeviceRelationId>
{
    LABEL "Relation type description";
    RELATION_TYPE CHILD_COMPONENT;
    ADDRESSING {<AddressVar>}
    COMPONENTS
    {
        <CommunicationDeviceComponentId>
        {
            AUTO_CREATE <autoCreate>;
            REQUIRED_RANGES
            {
                <AddressVar>{ MIN_VALUE <AddrMin>; MAX_VALUE <AddrMax>;}
            }
        }
    }
    MINIMUM_NUMBER <minNumber>;
    MAXIMUM_NUMBER <maxNumber>;
}
```

<DeviceComponentId>: The COMPONENT identifier identifies the component description for the device type.

<Label>: The string value shall contain a string that allows a human user to determine the function of the FDI Communication Server object.

<ValidateModules>: The Value refers to the METHOD implementing the module topology configuration validation function. Implementation details are specified in 5.2.9.

The attribute COMPONENT_RELATIONS allows to describe how modules can be connected. The definition of the COMPONENT_RELATIONS is optional. If used, it shall describe the relations to the CommunicationDevice definitions. The construct enables generic FDI Server-driven (device) topology configuration to be performed. Syntax details are described in IEC 61804-3. The subsequent text describes the semantic use of the COMPONENT_RELATION construct.

<CommunicationDeviceRelationId>: The attribute value identifies the COMPONENT_RELATION definition describing the relation between the device component and the CommunicationDevice component.

<CommunicationDeviceComponentId>: The attribute value has to match with a COMPONENT identifier used in a COMPONENT declaration that describes a CommunicationDevice (see 5.2.3).

<autoCreate>: The attribute value describes the number of CommunicationDevice components that can be automatically instantiated with the Device component.

<minNumber>/<maxNumber>/<autoCreate>: The attribute values define the instantiation constraints. The definition of these attributes is optional. The attribute values can contain conditional expressions.

The RELATION_TYPE shall be set to CHILD_COMPONENT.

<AddressVar>: The attribute value is a reference to a VARIABLE declaration. This VARIABLE holds the address value for a CommunicationDevice instance. The definition of this attribute is optional.

<AddrMin>/<AddrMax>: Values define the address value range for a CommunicationDevice instance. The value can for example correspond to a physical slot number. Usage of attributes ADDRESSING and REQUIRED_RANGES enables generic configuration routines.

5.2.3 CommunicationDevice component

Each FDI Communication Package shall contain at least one EDD element describing at least one CommunicationDevice component. A modular communication hardware structure shall be described by multiple CommunicationDevice COMPONENT descriptions:

```
COMPONENT <CommunicationDeviceComponentId>
{
    LABEL "<Label>";
    CAN_DELETE <CanDelete>;
    CLASSIFICATION NETWORK_COMPONENT;
    COMPONENT_RELATIONS
    {
        <CommunicationServiceProviderRelationId>
    }
}
```

```
COMPONENT_RELATION <CommunicationServiceProviderRelationId>
{
    LABEL "Relation between CommunicationDevice and communication service provider";
    RELATION_TYPE CHILD_COMPONENT;
    ADDRESSING {<AddressVar>}
    COMPONENTS
    {
        <CommunicationServiceProviderId>
        {
            AUTO_CREATE <autoCreate>;

```

```

    }
  }
  MINIMUM_NUMBER 1;
  MAXIMUM_NUMBER <maxNumber>;
}

```

<CommunicationDeviceComponentId>: The COMPONENT identifier identifies the CommunicationDevice component.

<Label>: The string value shall contain a human-readable string that allows a user to easily determine the function of the CommunicationDevice component.

<CanDelete>: Allowed values are TRUE or FALSE. It depends on whether a CommunicationDevice needs explicit configuration or whether the related communication service provider object shall be automatically instantiated with the CommunicationDevice. If the attribute CAN_DELETE is set to FALSE, the CommunicationDevice configuration is static.

The definition of the COMPONENT_RELATIONS is mandatory. It describes the relation to the communication service provider definition. The construct enables the FDI Server to instantiate communication service provider components in accordance with communication processing demands. Syntax details are described in IEC 61804-3. The subsequent text describes the semantic use of the COMPONENT_RELATION construct.

<CommunicationServiceProviderRelationId>: The attribute value identifies the COMPONENT_RELATION definition as such.

<CommunicationServiceProviderId>: The attribute value has to match with a COMPONENT identifier used in a COMPONENT declaration that describes a communication service provider (5.2.4).

<autoCreate>: The attribute value describes the number of communication service providers that can be automatically instantiated with the CommunicationDevice component.

The RELATION_TYPE shall be set to CHILD_COMPONENT.

The PROTOCOL attribute shall not be set.

5.2.4 Communication service provider component

Each FDI Communication Package describing a Communication Device shall contain at least one EDD element describing the communication service provider. The EDD component shall not define any configuration parameter.

```

COMPONENT <CommunicationServiceProviderId>
{
  LABEL "<Label>";
  BYTE_ORDER <byteOrder>;
  CAN_DELETE <CanDelete>;
  CLASSIFICATION NETWORK_COMMUNICATION_SERVICE_PROVIDER;
  COMPONENT_RELATIONS <CommunicationServiceProvidersConnectionPointRelationId>
  {
    <ConnectionPointRelationId>
  }
}

```

```

COMPONENT_RELATION <CommunicationServiceProvidersConnectionPointRelationId>
{
  LABEL "Relation between communication service provider and connection point";
  RELATION_TYPE CHILD_COMPONENT;
  ADDRESSING {<AddressVar>}
  COMPONENTS

```

```

{
    < ConnectionPointId>
    {
        AUTO_CREATE 1;
    }
}
MINIMUM_NUMBER 1;
MAXIMUM_NUMBER 1;
}

```

<CommunicationServiceProviderId>: The COMPONENT identifier identifies the communication service provider.

<Label>: The string value shall contain a human-readable string that allows a user to easily determine the function of the communication service provider object.

<CanDelete>: Allowed values are TRUE or FALSE. It depends on whether a communication service provider can be flexibly instantiated according to the communication processing demands. If the attribute CAN_DELETE is set to FALSE, the number of communication service provider component instantiations is static. The instantiation constraints declared through the attributes AUTO_CREATE, MINIMUM_NUMBER and MAXIMUM_NUMBER correspond to the capabilities of currently supported protocols.

<byteOrder>: The value enables generic integration of n-byte data types (e.g. 4-byte integer) into the communication message payload. The attribute value describes the byte order and shall be either BIG_ENDIAN or LITTLE_ENDIAN.

The definition of the COMPONENT_RELATIONS is mandatory. It describes the relation to the Connection Point definition. The construct enables to perform generic, FDI Server-driven topology configuration. Syntax details are described in IEC 61804-3. The subsequent text describes the semantic use of the COMPONENT_RELATION construct.

The Connection Point shall automatically be instantiated with the communication service provider and there shall be exactly one (1) Connection Point instance connected to the communication service provider. The instantiation constraints declared through the attributes AUTO_CREATE, MINIMUM_NUMBER and MAXIMUM_NUMBER correspond to the capabilities of currently supported protocols.

<CommunicationServiceProvidersConnectionPointRelationId>: the attribute value identifies the COMPONENT_RELATION declaration as such.

<ConnectionPointId>: the attribute value has to match with a COMPONENT identifier used for a COMPONENT declaration that describes a Connection Point (see 5.2.5).

The RELATION_TYPE shall be set to CHILD_COMPONENT.

The PROTOCOL attribute shall not be set.

5.2.5 Connection Point component

Each FDI Communication Package describing a Communication Device shall contain one EDD element describing one Connection Point for each of the protocols that are supported by the Communication Device:

```

COMPONENT <ConnectionPointId>
{
    LABEL "<Label>";
    CAN_DELETE FALSE;
    CLASSIFICATION NETWORK_CONNECTION_POINT;
}

```

```

    PROTOCOL <ProtocolId>;
    CONNECTION_POINT <ConnectionPointCollectionId>;
}

```

<ConnectionPointId>: the COMPONENT identifier identifies the Connection Point component declaration.

<Label>: the string value shall contain a string that allows a human user to determine the function of the Connection Point component.

<ProtocolId>: the value of this attribute indicates the communication capability, which allows the FDI Server to find other device types that can be connected to the network using the same type of protocol. For standardized protocols, the value is defined by the related fieldbus organization.

<ConnectionPointCollectionId>: the attribute value is a reference to a COLLECTION declaration that describes the data structure of the Connection Point as described in 5.2.6.

5.2.6 Connection Point collection

Each EDD describing the Connection Point of a communication device shall describe the COLLECTION element that describes the attributes that shall appear in the Information Model representation of the Connection Point. The protocol specific data exposed by the Connection Point identifies the device type and its network address.

```

COLLECTION <ConnectionPointCollectionId>
{
    LABEL "<Label>";
    MEMBERS
    {
        <AddressAttributeName>, <AddressAttributeVariableId>;
        VALID <VALID_VariableId>;
    }
}

```

<ConnectionPointCollectionId>: the identifier of the COLLECTION is referred by the CONNECTION_POINT attribute value defined in 7.7.3.5.

<Label>: the label identifies the Connection Point in a human readable way.

<AddressAttributeName>/<AddressAttributeVariableId>: the MEMBER section refers to the VARIABLE definitions describing the address attributes implemented by a Connection Point. The content of the MEMBER section is protocol-specific.

<VALID>/<VALID_VariableId> is a Collection member referring a Boolean VARIABLE holding the validation status that shall be set by the ValidateNetwork Action (see 5.2.8).

5.2.7 Network component

Each FDI Communication Package describing a Communication Device shall contain one EDD element describing one Network for each of the protocols that are supported by the Communication Device. The definition supports the network topology engineering:

```

COMPONENT <NetworkComponentId>
{
    LABEL "<Label>";
    CAN_DELETE TRUE;
    CHECK_CONFIGURATION <Validate>;
    CLASSIFICATION NETWORK;
    PROTOCOL <ProtocolId>;
    COMPONENT_RELATIONS
    {
        <NetworksConnectionPointRelationId>
    }
}

```

```

    }
}

COMPONENT_RELATION <NetworksConnectionPointRelationId>
{
    LABEL "Relation between network and connection point";
    RELATION_TYPE CHILD_COMPONENT;
    ADDRESSING {<AddressVar>}
    COMPONENTS
    {
        <ConnectionPointId>
        {
            REQUIRED_RANGES
            {
                <BusAddressVar>{ MIN_VALUE <BusAddrMin>; MAX_VALUE <BusAddrMax>;}
            }
        }
    }
    MINIMUM_NUMBER 1;
    MAXIMUM_NUMBER <maxNumber>;
}

```

<NetworkComponentId>: the COMPONENT identifier identifies the Network component declaration.

<Label>: the string value shall contain a human-readable string that allows a user to easily determine the function of the Network component.

<Validate>: the value refers to the METHOD implementing the network topology configuration validation function (see 5.2.8).

<ProtocolID>: the value of this attribute allows the FDI Server to find other device types that can be connected to the network using the same type of protocol. For standardized protocols, the value is defined by the related fieldbus organization.

The definition of the COMPONENT_RELATIONS is mandatory. It describes the relation to the Connection Point definition and by that the capabilities of a network. The construct enables generic FDI Server-driven network topology configuration to be performed. Syntax details are described in IEC 61804-3. The subsequent text describes the semantic use of the COMPONENT_RELATION construct.

<NetworksConnectionPointRelationId>: the attribute value identifies the COMPONENT_RELATION definition.

<ConnectionPointId>: The attribute value has to match with a COMPONENT identifier used for a COMPONENT declaration that describes a Connection Point (see 5.2.4).

<maxNumber>: The attribute value limits the number of Connection Points that can be connected to the network. The attribute values can contain conditional expressions.

The RELATION_TYPE shall be set to CHILD_COMPONENT.

<BusAddressVar>: The attribute value is a reference to a VARIABLE declaration. This VARIABLE holds the network address value for any device that is connected to the network.

<BusAddrMin>/<BusAddrMax>: Values define the network address value range.

5.2.8 ValidateNetwork

The method ValidateNetwork represents the Communication Device implemented Business Logic that validates a current network topology. The ValidateNetwork method handles any necessary dependencies related to bus parameters. The implementation of related EDDL logic is based on the EDDL Built-in function ObjectReference, which enables the analysis of a set of child instances (Connection Point instances). The validation logic shall set the <VALID> attribute of the Connection Point instance that has passed the validation.

The implementation of ValidateModules is optional if the module setup is either static or if the configuration rules defined in the COMPONENT construct are sufficient to configure the module setup.

Table 1 shows the ValidateNetwork Action Arguments.

Signature

```
ValidateNetwork (
    [out] Integer ServiceError,
    [out] String ErrorMessage);
```

Table 1 – ValidateNetwork Action arguments

Argument	Description
ServiceError	0: OK -1: Failed / the Connection Point that did not pass the validation is indicated by the <VALID> attribute () value set to false. Remark: The argument values correspond to the error codes specified in IEC 61804-3 named BI_SUCCESS (value = 0) and BI_ERROR (value = -1). The Action returns the ServiceError result using the "return" statement.
ErrorMessage	If the method returns an empty string (NULL), the Action call succeeded. In the case of an error, the Action can return a description of the problem.

5.2.9 ValidateModules

The method ValidateModules validates the current module setup. The implementation of the related EDDL logic is based on the EDDL Built-in function ObjectReference, which enables a set of child instances to be analyzed. The implementation of ValidateModules is optional if the module setup is either static or if the configuration rules defined in the COMPONENT construct are sufficient to configure the module setup.

NOTE The decision whether ValidateModules is needed or not is vendor-specific.

Table 2 shows the ValidateModules Action Arguments.

Signature

```
ValidateModules (  
    [out] Integer serviceError,  
    [out] String ErrorMessage);
```

Table 2 – ValidateModules Action arguments

Argument	Description
ServiceError	0: OK -1: Failed / the Connection Point that did not pass the validation is indicated by the <VALID> attribute () value set to false. Remark: The Argument values correspond to the error codes specified in IEC 61804-3 named BI_SUCCESS (value = 0) and BI_ERROR (value = -1). The Action returns the ServiceError result using the "return" statement.
ErrorMessage	If the Action returns an empty string (NULL), the method call succeeded. In the case of an error, the Action can return a problem description.

5.2.10 UIP specifics

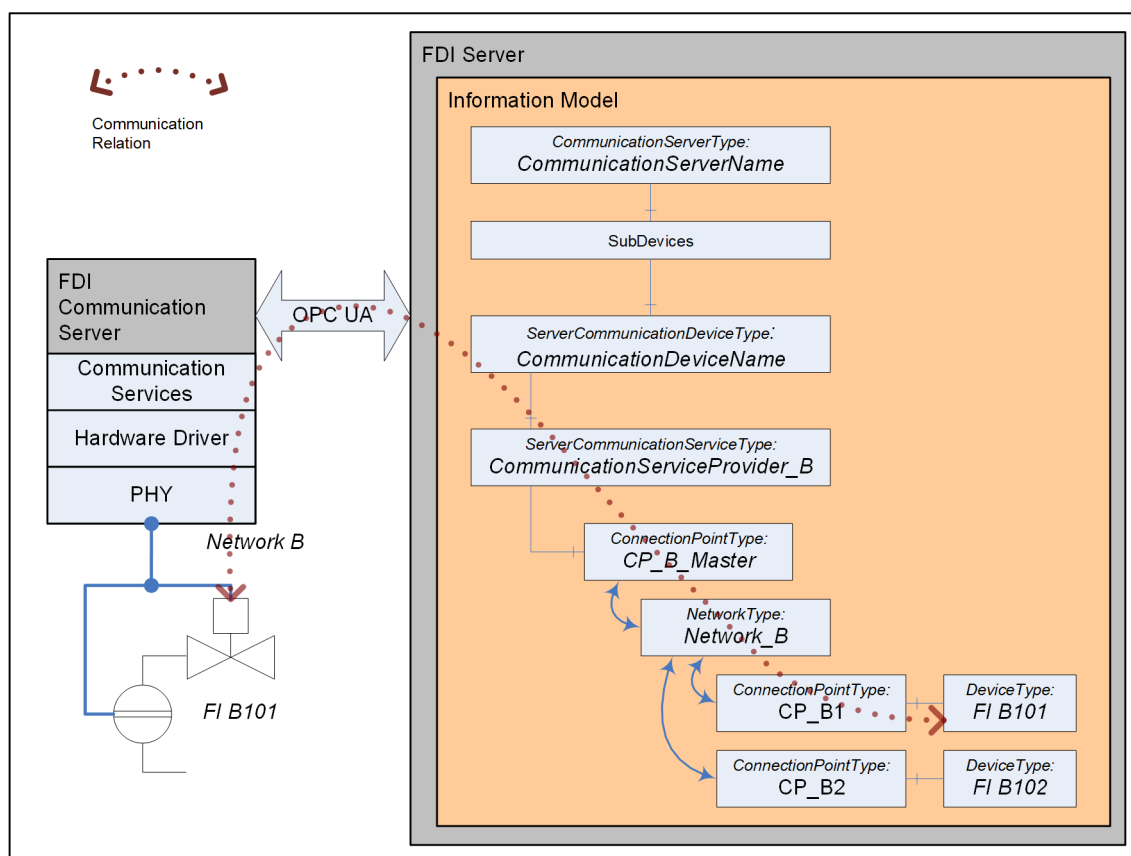
The FDI Communication Package can contain the UIP to support, for example, diagnostics and parameterization.

5.2.11 Deployment

The FDI Server imports the FDI Communication Package. The handling of EDD and UIP parts matches with the import procedure performed for the FDI Package (see IEC 62769-2 and IEC 62769-3).

6 Communication relations

The purpose of a communication device and its communication services is to exchange information between the physical device and the device representation managed by the FDI Server. The information exchange is managed via communication relations, see Figure 3. An established communication relation represents the capability to exchange information between the FDI Server-managed device representation and the physical device. The use of a Communication Relation allows abstracting from protocol specifics typically used to manage connections.



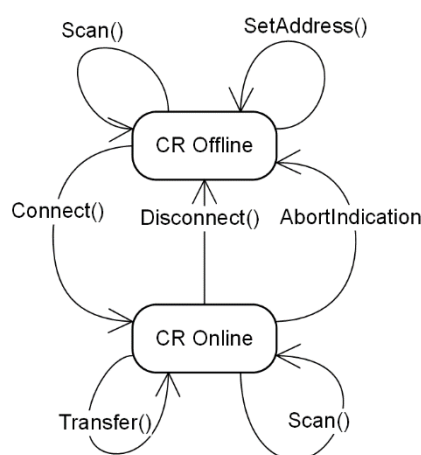
IEC

Figure 3 – Communication relation

NOTE 1 The core of information exchange happens between the device connected to the physical network and the corresponding instance within the Information Model but does not cover the complete device application.

The following state chart (Figure 4) describes the general state flow for a single communication relation. The diagram also shows which communication services can be invoked during a "CR Online" state.

The "AbortIndication" shown in Figure 4 can be detected in different protocol-specific ways. The one specified for any communication device is bound to the serviceErrors returned by the specified communication services. Even the Scan method can determine a connection loss, when the device for which a communication relation has been activated does not appear in a scan result.



IEC

Figure 4 – Communication relation state chart

NOTE 2 The management of communication relations is optional.

7 FDI Communication Server definition

7.1 General

In terms of FDI, the FDI Communication Server is a dedicated OPC UA Server providing access to one or more field device networks. Each FDI Communication Server is modelled as a Modular Device where each module (also called CommunicationDevice in the sequence) represents the access point to one network.

The Modular Device itself represents the FDI Communication Server as a whole.

7.2 General characteristics

The FDI Communication Server implements characteristics for each of its CommunicationDevices specified in 7.3.3. Additionally, an FDI Communication Server implements the following characteristics:

- The FDI Server always synchronizes (see 7.5, 7.8.8, and 7.8.11) the FDI Communication Server hosted Information Model from the FDI Server hosted Information Model content.
- CommunicationDevices can be statically instantiated or they can be created/deleted by the FDI Server.
- Communication between the FDI Server and the FDI Communication Server is based on OPC UA. OPC UA specifies a wire protocol for its services that can be implemented on arbitrary platforms and runtime environments.
- To avoid race conditions, the FDI Communication Server only allows one FDI Server being connected at a time. With this restriction an FDI Communication Server can refrain from any synchronization (locking) mechanism. The FDI specification does not enforce FDI Communication Servers implementing any interlocking mechanism to manage concurrent access to a single device connected to the physical network.

7.3 Information Model

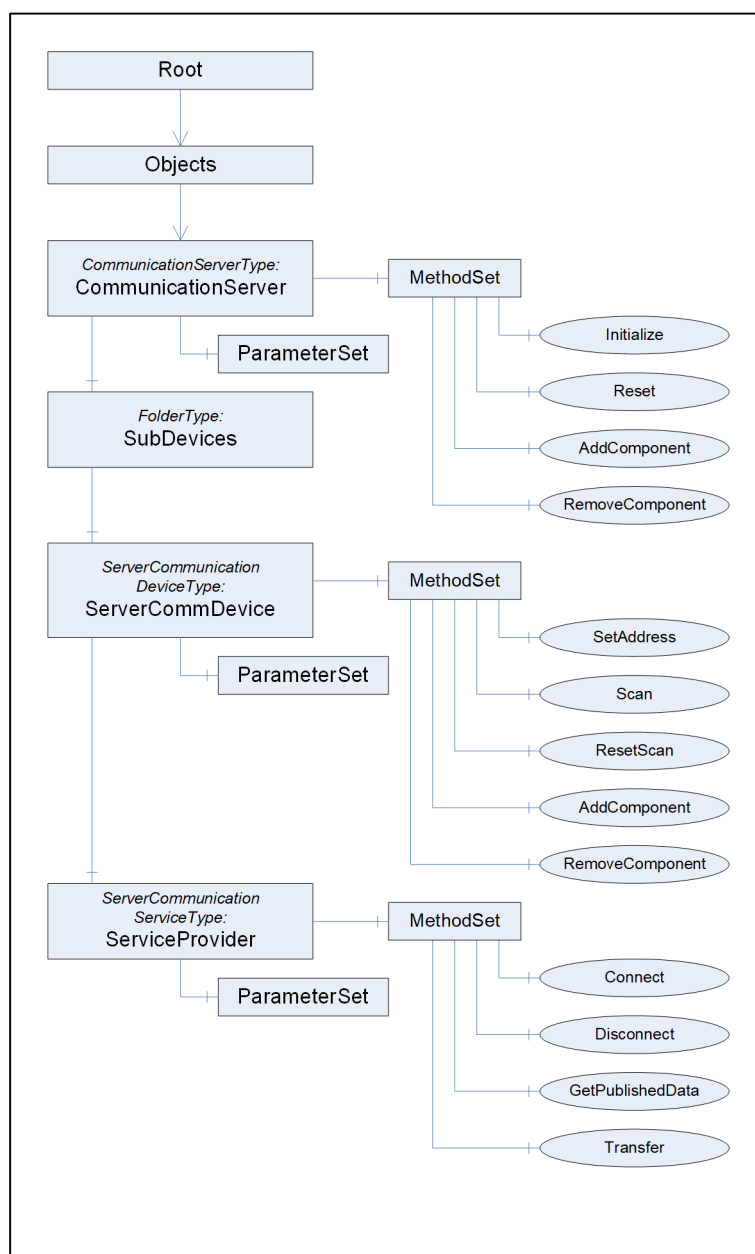
7.3.1 General

Subclause 7.3 specifies the FDI Communication Server hosted Information Model.

An FDI Communication Server is an OPC UA Server that encapsulates communication hardware and provides standardized communication ability. The FDI Server connects to the FDI Communication Server as an OPC UA Client and accesses the networks supported by the FDI Communication Server via the FDI Communication Server information model. The task of the FDI Communication Server is to expose this information model. The FDI Communication Server shall not maintain Device Instances or network topology information. All interaction with FDI devices is done through the FDI Server and just transferred by the FDI Communication Server.

For the FDI Server, an FDI Communication Server looks like a device that supports FDI Communication Services and uses OPC UA to communicate. The FDI Communication Server may run locally on the same PC as the FDI Server (loop-back adapter) or remote in the field (e.g. embedded into a controller). Like a device, each FDI Communication Server has an associated FDI Package. This FDI Package is used to create communication devices in the Information Model of the FDI Server that represent access to the networks implemented by the FDI Communication Server.

The Information Model of an FDI Communication Server is based on the Information Model defined in IEC 62769-5. Figure 5 replicates the Modular Device structure and illustrates how it maps into the overall AddressSpace. The modules represent the communication channels of the FDI Communication Server.



IEC

Figure 5 – FDI Communication Server AddressSpace

The CommunicationServerType (the root of the Modular Device) is a subtype of the DeviceType. The MethodSet contains the methods Initialize, Reset, AddComponent and RemoveComponent. The methods AddComponent and RemoveComponent are optionally present if the FDI Communication Server supports the dynamic instantiation of elements in the folder SubDevices.

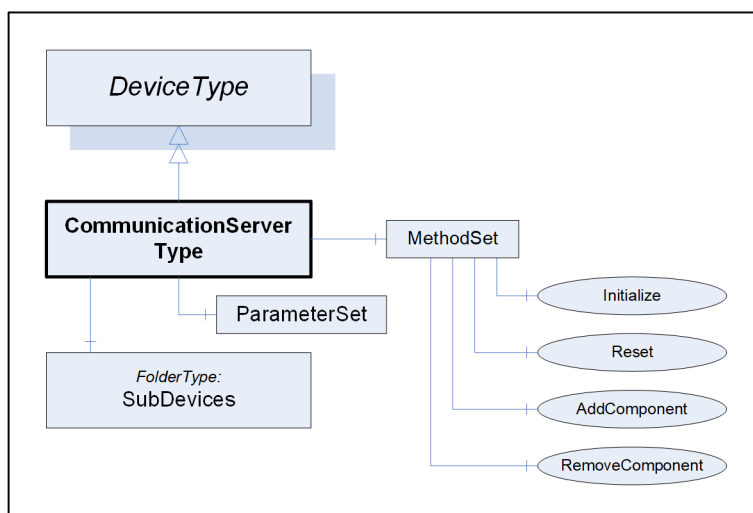
All sub-devices are instances of the ServerCommunicationDeviceType defined in 7.3.3. The instances of the ServerCommunicationDeviceType (ServerCommDevice) have a MethodSet that can implement the methods SetAddress, Scan, AddComponent, RemoveComponent. AddComponent and RemoveComponent are optionally present if the FDI Communication Server supports a variable number of instances of the ServerCommunicationServiceType.

Formal definitions are found in 7.3.2, 7.3.3 and 7.3.4.

7.3.2 CommunicationServerType

7.3.2.1 General

The CommunicationServerType is a subtype of the DeviceType and provides the methods needed to manage the instances ServerCommunicationDeviceType. Figure 6 shows the CommunicationServerType definition that is formally defined in Table 3 and Table 4.



IEC

Figure 6 – CommunicationServerType

Table 3 – CommunicationServerType definition

Attribute	Value				
BrowseName	CommunicationServerType				
IsAbstract	False				
References	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of the DeviceType defined in IEC 62541-100.					
HasComponent	Object	MethodSet		BaseObjectType	Mandatory
HasComponent	Object	ParameterSet		BaseObjectType	Optional
HasComponent	Object	SubDevices		FolderType	Mandatory

Table 4 – MethodSet of CommunicationServerType

Attribute	Value				
BrowseName	MethodSet				
IsAbstract	False				
References	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
HasComponent	Method	Initialize			Mandatory
HasComponent	Method	Reset			Mandatory
HasComponent	Method	AddComponent			Optional
HasComponent	Method	RemoveComponent			Optional

The CommunicationServerType and each instance of this Type share the same Methods. The NodeId of these Methods will be fixed and defined in this document (see Annex B). FDI Communication Server clients therefore do not have to browse for these Methods. They can use the fixed NodeId as the MethodId of the Call Service.

The additional Methods AddComponent and RemoveComponent add the ability to add or remove instances of ServerCommunicationDeviceType according to the communication hardware's structure. These services are not applicable if the FDI Communication Server implements a static communication hardware structure.

The SubDevices folder contains instances of ServerCommunicationDeviceType that represent the communication modules.

NOTE The indication for a static communication hardware layout is indicated in the FDI Package with COMPONENT attribute CAN_DELETE set to FALSE in COMPONENT declarations.

7.3.2.2 Reset Method

Reset is used to reset the communication hardware and related driver software. Any ongoing communication will be stopped immediately. All communication channels enter the closed status.

The Method Reset shall not be present in the FDI Server-hosted Information Model. The FDI Server shall be able to handle the shut-down procedure automatically according to communication demands.

Typically, the FDI Communication Server operation includes some hardware and protocol driver handling that can be independent from any modular structure. Because of this possibility, the Reset method is arranged underneath the CommunicationServerType. For the purpose of reducing the complexity of FDI Communication Server operation, only one Reset method has been specified.

The signature of this Method is specified below. Table 5 and Table 6 specify the arguments and AddressSpace representation, respectively.

Signature

```
Reset (
    [out] Integer      serviceError);
```

Table 5 – Reset Method arguments

Argument	Description
serviceError	0: OK -1: Failed

Table 6 – Reset Method AddressSpace definition

Attribute	Value				
BrowseName	Reset				
References	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
HasProperty	Variable	OutputArguments	Argument[]	PropertyType	Mandatory

7.3.2.3 Initialize Method

Initialize is used to initialize the communication hardware. The initialization function of the FDI Communication Server shall use the parameterization data hosted by the ParameterSet that is contained within the instance of the CommunicationServerType and all instances of ServerCommunicationDeviceType.

In order to enable parameter changes during operation, the Initialize method can be re-invoked. If the FDI Communication Server needs to reset its communication hardware, it shall automatically restore any communication relation that existed. A modular FDI Communication Server can flexibly initialize only those ServerCommunicationDeviceType instances for which configuration changes have been detected.

The Method Initialize shall not be present in the FDI Server-hosted Information Model. The FDI Server shall be able to handle the start procedure automatically according to human-driven communication requests.

The FDI Communication Server operation can include some hardware and protocol driver handling that can be independent from any modular structure. Because of this possibility, the Initialize method is arranged underneath the CommunicationServerType. For the purpose of reducing the complexity of the FDI Communication Server's operation, only one Initialize method has been specified.

The signature of this Method is specified below. Table 7 and Table 8 specify the arguments and AddressSpace representation, respectively.

Signature

```
Initialize(
    [out] Integer      serviceError)
```

Table 7 – Initialize Method arguments

Argument	Description
serviceError	0: OK -1: Failed

Table 8 – Initialize Method AddressSpace definition

Attribute	Value				
BrowseName	Initialize				
References	NodeClass	BrowseName	Data Type	TypeDefinition	ModellingRule
HasProperty	Variable	OutputArguments	Argument[]	PropertyType	Mandatory

7.3.2.4 AddComponent Method

AddComponent shall be used to configure the modular setup of an FDI Communication Server when the FDI Communication Server has no statically defined communication hardware setup. This method shall be used to add a module (Instance of ServerCommunicationDeviceType).

The signature of this Method is specified below. Table 9 and Table 10 specify the arguments and AddressSpace representation, respectively.

Signature

```

AddComponent (
    [in] String      ModuleTypeName,
    [in] String      InstanceName,
    [in] String      InstanceLabel,
    [out] NodeId     InstanceNodeId,
    [out] Integer    ServiceError);

```

Table 9 – AddComponent Method arguments

Argument	Description
ModuleTypeName	Type of module to be created as defined in the FDI Package. The module type name shall correspond to one of the COMPONENT identifier definitions (see 5.2.3).
InstanceName	Non-localized name of the module's Device Node of the created element. This name has to be unique within the scope of the FDI Communication Server's Information Model.
InstanceLabel	Human-readable label for the root Node of the created module.
InstanceNodeId	Callee-assigned identifier for the module's Device Node.
ServiceError	0 – OK -1 – E_InvalidType – a module for the specified Type can not (not anymore) be added -2 – E_DuplicateName – there exists already a module with the same name as specified with the InstanceName argument -3 – E_UnknownType – an unknown ModuleTypeName has been specified -4 – E_LimitExceeded – the total number of modules is exceeded (this might be caused by power constraints or other resource limitations)

Table 10 – AddComponent Method AddressSpace definition

Attribute	Value				
BrowseName	AddComponent				
References	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
HasProperty	Variable	InputArguments	Argument[]	PropertyType	Mandatory
HasProperty	Variable	OutputArguments	Argument[]	PropertyType	Mandatory

7.3.2.5 RemoveComponent Method

RemoveComponent shall be used to remove a module (Instance of ServerCommunicationDeviceType). Implementation of RemoveComponent is optional if the communication hardware setup is static.

The signature of this Method is specified below. Table 11 and Table 12 specify the arguments and AddressSpace representation, respectively.

Signature

```
RemoveComponent (
    [in]  NodeId      ModuleNodeId,
    [out] Integer     ServiceError);
```

Table 11 – RemoveComponent Method arguments

Argument	Description
ModuleNodeId	The value is the identification of the existing instance in the Information Model.
ServiceError	0: OK -1: Failed, the specified node does not exist

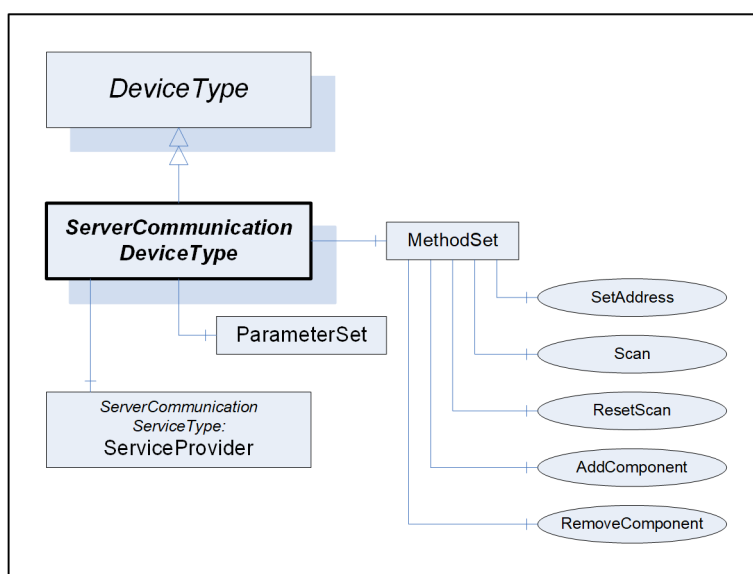
Table 12 – RemoveComponent Method AddressSpace definition

Attribute	Value				
BrowseName	RemoveComponent				
References	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
HasProperty	Variable	InputArguments	Argument[]	PropertyType	Mandatory
HasProperty	Variable	OutputArguments	Argument[]	PropertyType	Mandatory

7.3.3 ServerCommunicationDeviceType

7.3.3.1 General

The ServerCommunicationDeviceType represents a communication channel for a particular network. The ServerCommunicationDeviceType is a subtype of the DeviceType. The ParameterSet for each instance of a ServerCommunicationDevice will contain Parameters necessary to configure the operation of the network. Figure 7 shows the ServerCommunicationDeviceType definition that is formally defined in Table 13 and Figure 14.



IEC

Figure 7 – ServerCommunicationDeviceType

Table 13 – ServerCommunicationDeviceType definition

Attribute	Value				
BrowseName	ServerCommunicationDeviceType				
IsAbstract	True				
References	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of the DeviceType defined in OPC UA Part DI.					
HasComponent	Object	MethodSet		BaseObjectType	Optional
HasComponent	Object	ParameterSet		BaseObjectType	Optional
HasComponent	Object	ServiceProvider		ServerCommunicationServiceType	Mandatory
HasProperty	Variable	ListOfCommunicationProfiles	String[]	PropertyType	Mandatory

The Property ListOfCommunicationProfiles contains a list of communication profiles supported by the ServerCommunicationDevice. Valid strings are defined in IEC 62769-4:2020, Annex F.

Table 14 – MethodSet of ServerCommunicationDeviceType

Attribute	Value				
BrowseName	MethodSet				
IsAbstract	True				
References	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
HasComponent	Method	Scan			Optional
HasComponent	Method	ResetScan			Optional
HasComponent	Method	SetAddress			Optional
HasComponent	Method	AddComponent			Optional
HasComponent	Method	RemoveComponent			Optional

7.3.3.2 Scan Method

Scan shall be used to start discovering devices connected to the physical network. The associations between the method Scan and the corresponding physical network connection enables the FDI Communication Server to access the correct physical network connection. The Scan method is implemented by the Communication Server runtime module.

The signature of this Method is specified below. Table 15 and Table 16 specify the arguments and AddressSpace representation, respectively.

NOTE 1 Communication Servers can run the network scan in a background task, so the invocation of the function Scan will return cached network scan results.

NOTE 2 If the SCAN takes very long, the FDI Communication Server might return an empty TopologyScanResult and the ServiceError 1 identifying that the scan is still running.

Signature

```
Scan (
    [out] XmlElement      TopologyScanResult,
    [out] Integer         ServiceError)
```

Table 15 – Scan Method arguments

Argument	Description
TopologyScanResult	The argument value is an XML formatted string representing a list of devices connected to the physical network. Each of the devices connected to the physical network is represented by a data structure matching with a Connection Point node. Connection Point attributes are protocol-specific. The corresponding topologyScanResult schema is specified in IEC 62769-4:2020, Annex F. Returns an empty string for TopologyScanResult in the case of any error.
ServiceError	0: OK / scan completed 1: OK / get complete scan result by calling Scan again -1: Failed / not initialized -2: Failed / not connected to a network -3: Failed / no device found, the topologyScanResult is empty

Table 16 – Scan Method AddressSpace definition

Attribute	Value				
BrowseName	Scan				
References	NodeClass	BrowseName	Data Type	TypeDefinition	ModellingRule
HasProperty	Variable	OutputArguments	Argument[]	PropertyType	Mandatory

7.3.3.3 ResetScan Method

ResetScan shall be used to reset the internal cache of scan results. It will also cancel a running scan if the FDI Communication Server scan mechanism supports this.

The signature of this Method is specified below. Table 17 and Table 18 specify the arguments and AddressSpace representation, respectively.

Signature

```
ResetScan (
    [out] Integer         ServiceError)
```

Table 17 – ResetScan Method arguments

Argument	Description
ServiceError	0: OK / scan reset -1: Failed / not initialized -2: Failed / not connected to a network

Table 18 – ResetScan Method AddressSpace definition

Attribute	Value				
BrowseName	ResetScan				
References	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
HasProperty	Variable	OutputArguments	Argument[]	PropertyType	Mandatory

7.3.3.4 SetAddress Method

SetAddress shall be used to change the network address (communication address) of a device. The Communication Device shall ensure unique network address values. If the argument value of newAddress is already assigned to a device connected to the physical network, the Communication Device shall return the argument serviceError value "-4: Failed/duplicate Address error".

It depends on the protocol whether the address assignment service shall work even when a communication relation is already established.

The signature of this Method is specified below. The arguments for SetAddress Method are described in Table 19.

Signature

```
SetAddress (
    [in]                <OldAddress>,
    [in]                <NewAddress>,
    [out] Int32         ServiceError);
```

Table 19 – SetAddress Method arguments

Argument	Description
<OldAddress>	This is a placeholder for a 1..n protocol-specific arguments representing the existing protocol-specific network address of the device connected to the physical network. Values that represent a network address are specified in IEC 62769-4:2020, Annex F.
<NewAddress>	This is a placeholder for 1..n protocol-specific arguments representing the new protocol-specific network address that shall be assigned to a device connected to the physical network. Values that represent a network address's arguments are specified in IEC 62769-4:2020, Annex F.
ServiceError	0: OK/execution finished successfully Other values that represent ServiceErrors are specified in IEC 62769-4:2020, Annex F.

7.3.4 ServerCommunicationServiceType

7.3.4.1 General

Communication services provide the means to communicate with a Device or to, for example, execute a Scan on a Network. Communication services are represented through Methods in the Information Model (see IEC 62769-5).

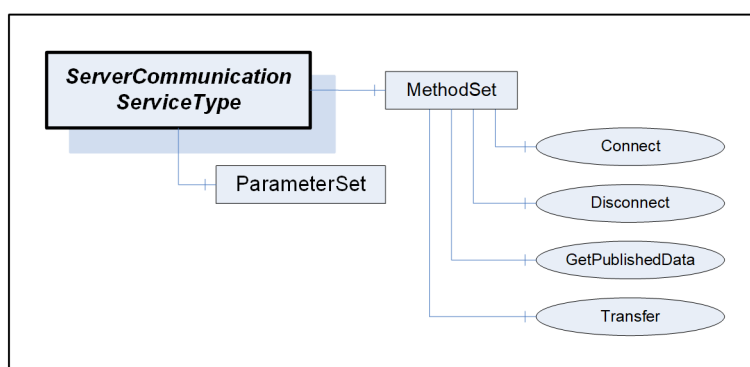
The formal definition of ServerCommunicationServiceType is found in Figure 8, Table 20 and Table 21.

The NodeId of these Methods will be fixed and defined in this document. FDI Clients therefore do not have to browse for these Methods. They can use the fixed NodeId as the MethodId of the Call Service.

Communication methods that include their NodeIds are uniquely defined in this document. FDI Clients can use the Methods directly (without browsing). The OPC UA Call Service shall be used as follows:

- the MethodId argument shall contain the fixed NodeId of the Method;
- the ObjectId argument shall contain the NodeId of the MethodSet.

The OPC UA StatusCode Bad_MethodInvalid shall be returned from the Call Service for elements where the communication methods are not supported.



IEC

Figure 8 – ServerCommunicationServiceType

Table 20 – ServerCommunicationServiceType definition

Attribute	Value				
BrowseName	ServerCommunicationServiceType				
IsAbstract					
References	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
Subtype of the DeviceType defined in OPC UA Part DI.					
HasComponent	Object	MethodSet		BaseObjectType	Mandatory
HasComponent	Object	ParameterSet		BaseObjectType	Optional

Table 21 – MethodSet of ServerCommunicationServiceType

Attribute	Value				
BrowseName	MethodSet				
IsAbstract					
References	NodeClass	BrowseName	DataType	TypeDefinition	ModellingRule
HasComponent	Method	Connect			Optional
HasComponent	Method	Disconnect			Optional
HasComponent	Method	Transfer			Mandatory
HasComponent	Method	GetPublishedData			Optional

7.3.4.2 Connect Method

Connect shall be used to establish a communication relation to a device that is physically connected to the Network. Establishing the communication relation may imply checks of identification data that are part of the addressData with data inside the physical device. The Communication Device performs this DeviceType match verification in accordance with a corresponding network protocol standard. Related details are specified in IEC 62769-4:2020, Annex F.

The devices' address is contained in the Connection Point of the corresponding Device Instance within the Information Model (Device Connection Point). The communication relation between the Information Model-associated device application and the physical device is further on identified by the communication relation identifier. Details about how to manage the status of a communication relation is described in Clause 6. Connection Point descriptions may be layered as described in Annex A.

NOTE 1 As the NodeId is a unique identifier within the Information Model scope, the NodeId of the Device Connection Point can be a unique identifier for any communication relation in the scope of a communication device.

NOTE 2 The term communication relation is introduced to describe the status of an infrastructure that enables data exchange between Information Model hosted data and a physical device. If the communication relation is established, data exchange is possible.

The signature of this Method is specified below. Table 22 specifies the arguments.

Signature

```
Connect (
    [in]   ByteString      CommunicationRelationId,
    [in]   <AddressData>,
    [out]  <DeviceInformation>,
    [out]  Int32           ServiceError);
```

Table 22 – Connect Method arguments

Argument	Description
CommunicationRelationId	This is a client generated ID that is used to uniquely identify this connection. This could be an index (e.g. a NodeId) that the client (= FDI Server) needs in order to identify entries in its topology.
<AddressData>	This is a placeholder for a protocol-specific argument list that is used for the address and optional device identification data (details described in IEC 62769-4:2020, Annex F).
<DeviceInformation>	This is a placeholder for a protocol-specific argument list in which the connect result data are stored.
ServiceError	0: OK/execution finished, connection established successfully Other values that represent ServiceErrors are specified in IEC 62769-4:2020, Annex F.

7.3.4.3 Disconnect method

Disconnect shall be used to terminate a communication relation to a Device.

The signature of this Method is specified below. Attributes of the Disconnect method are specified in Table 23. Disconnect is a synchronous method call.

Signature

```
Disconnect (
    [in]   ByteString      CommunicationRelationId,
    [out]  Int32           ServiceError);
```

Table 23 – Disconnect Method arguments

Argument	Description
CommunicationRelationId	Same ID as used in method Connect specified in 7.3.4.2.
ServiceError	1: OK/disconnect finished successfully Other values that represent ServiceErrors are specified in IEC 62769-4:2020, Annex F.

7.3.4.4 Transfer method

Transfer shall be used to perform information exchange with a Device.

The signature of this Method is specified below. All arguments are specified in Table 24.

Signature

```
Transfer (
    [in]   ByteString      CommunicationRelationId,
    [in]                               <SendData>,
    [out]                               <ReceiveData>,
    [out]  Int32           ServiceError);
```

Table 24 – Transfer Method arguments

Argument	Description
CommunicationRelationId	See 7.3.4.2.
<SendData>	This is a placeholder for a protocol-specific list of arguments as described in IEC 62769-4:2020, Annex F. The argument values represent the protocol-specific communication service request that is sent to the device.
<ReceiveData>	This is a placeholder for a protocol-specific list of arguments as described in IEC 62769-4:2020, Annex F. The argument values represent the protocol-specific communication service response that is received from the device.
ServiceError	0: OK / execution finished, ReceivedData contains the result Other values that represent ServiceErrors are specified in IEC 62769-4:2020, Annex F.

7.3.4.5 GetPublishedData Method

The FDI Server sends GetPublishedData requests to the FDI Communication Server to receive data that is submitted by unsolicited data messages. The data contained in the argument SendData prepares the exchange of "unsolicited" data messages from the device. The content of SendData is protocol-specific. The FDI Communication Server queues GetPublishedData requests in a queue associated with the Communication Relation defined through the argument CommunicationRelationId. The argument PublishId identifies the related queue entry. Each time the FDI Communication Server receives unsolicited data messages, it saves the received data in association with the existing queue entry that has been created for the GetPublishedData. Depending on the underlying network technology (performance), the method GetPublishedData can immediately return with data coming from an "unsolicited" data message.

Subsequent pulling of data that is submitted by unsolicited data messages works through the same method `GetPublishedData`. In this case, the argument `SendData` is empty. The argument `PublishId` matches with the value that has been provided with the initial call `GetPublishedData` that has established the transmission of exchange of "unsolicited" data messages.

In order to stop the device sending the "unsolicited" data, the method `GetPublishedData` shall be used again but the argument `SendData` contained data terminates the exchange of "unsolicited" data messages from the device. Table 25 shows the `GetPublishedData` Method Arguments.

Signature

```
GetPublishedData (
    [in]   ByteString      CommunicationRelationId,
    [in]   <SendData>,
    [out]  <ReceiveData>,
    [out]  DateTime        TimeStamp
    [in]   UInt32          PublishId,
    [out]  Int32            ServiceError);
```

Table 25 – GetPublishedData Method arguments

Argument	Description
CommunicationRelationId	See 7.3.4.2.
<SendData>	This is a placeholder for a protocol-specific list of arguments as described in IEC 62769-4:2020, Annex F. The argument values control the exchange of unsolicited messages.
<ReceiveData>	This is a placeholder for a protocol-specific list of arguments as described in IEC 62769-4:2020, Annex F. The argument values convey data that comes from unsolicited messages.
TimeStamp	Time when the data was published by the device.
PublishId	The number identifies an established subscription that conveys data that comes from unsolicited messages.
ServiceError	0: OK / execution finished Other values that represent ServiceErrors are specified in IEC 62769-4:2020, Annex F.

7.4 OPC UA Server Profile for FDI Communication Server

Profiles are named groupings of `ConformanceUnits` as defined in IEC 62541-6 and IEC 62541-7. The term "Facet" in the title of a Profile indicates that this Profile is expected to be part of another larger Profile or concerns a specific aspect of OPC UA. Profiles with the term "Facet" in their title are expected to be combined with other Profiles to define the complete functionality of an OPC UA Server or Client. The minimum required OPC UA Server Profile is the "Micro Embedded Device Server Profile".

The following table specifies the facet for an OPC UA Server that acts as an FDI Communication Server. Table 26 describes `Conformance Units` included in this facet.

Table 26 – FDICommunicationServer_Facet definition

Conformance Unit	Description	Optional/Mandatory
FDI Communication Server Information Model	Support at least one instance of <code>CommunicationServerType</code> .	M