



DPC: 25 / 30509891 DC

BSI Group Headquarters

389 Chiswick High Road London W4 4AL

Tel: +44 (0)20 8996 9000

Fax: +44 (0)20 8996 7400

www.bsigroup.com

Date: 07 February 2025

Origin: International

Latest date for receipt of comments: 26 February 2025

Project No. 2025/00431

Responsible committee: GEL/65 Measurement and control Interested committees:

Title: Draft BS EN 63278-5 Asset Administration Shell for industrial applications

Part 5: Interfaces

Please notify the secretary if you are aware of any keywords that might assist in classifying or identifying the standard or if the content of this standard

- i) has any issues related to 3rd party IPR, patent or copyright
- ii) affects other national standard(s)
- iii) requires additional national guidance or information

**WARNING: THIS IS A DRAFT AND MUST NOT BE REGARDED OR USED AS A BRITISH STANDARD.
THIS DRAFT IS NOT CURRENT BEYOND 26 February 2025**

This draft is issued to allow comments from interested parties; all comments will be given consideration prior to publication. No acknowledgement will normally be sent. **See overleaf for information on the submission of comments.**

No copying is allowed, in any form, without prior written permission from BSI except as permitted under the Copyright, Designs and Patent Act 1988 or for circulation within a nominating organization for briefing purposes. Electronic circulation is limited to dissemination by e-mail within such an organization by committee members.

Further copies of this draft may be purchased from BSI Shop <http://shop.bsigroup.com> or from BSI Customer Services, Tel: +44(0) 20 8996 9001 or email cservices@bsigroup.com. British, International and foreign standards are also available from BSI Customer Services.

Information on the co-operating organizations represented on the committees referenced above may be obtained from <http://standardsdevelopment.bsigroup.com>

Responsible Committee Secretary: **Ms Charmaine Mukherjee (BSI)**

Direct tel:

E-mail: charmaine.mukherjee@bsigroup.com

Introduction

This draft standard is based on national and international discussions. Your comments on this draft are invited and will assist in the preparation of the consequent standard.

For international standards, comments will be reviewed by the relevant UK national committee before sending the consensus UK vote and comments to the international committee, which will then decide appropriate action. If the international standard is approved, it is usual for the text to be published as a British Standard.

For national standards, comments will be reviewed by the relevant UK national committee and the resulting standards published as a British Standard.

UK Vote

Please indicate whether you consider the UK should submit a negative (with supporting technical reasons) or positive vote on this draft. Please indicate if you are aware of any reason why this draft standard should not be published as a British Standard.

Submission of Comments

- **Annotated drafts are not acceptable and will be rejected.**
- All comments should be submitted online at <https://standardsdevelopment.bsigroup.com/>. You will need to register in order to comment.

Template for comments and secretariat observations

Date: xx/xx/20xx	Document: ISO/DIS xxxx
------------------	------------------------

1	2	(3)	4	5	(6)	7
MB	Clause No./ Subclause No./Annex (e.g. 3.1)	Paragraph/Figure/ Table/Note	Type of comment	Commend (justification for change) by the MB	Proposed change by the MB	Secretariat observations on each comment submitted
	3.1	Definition 1	ed	Definition is ambiguous and needs clarifying.	Amend to read '...so that the mains connector to which no connection...'	
	6.4	Paragraph 2	te	The use of the UV photometer as an alternative cannot be supported as serious problems have been encountered in its use in the UK.	Delete reference to UV photometer.	



65/1115/CD

COMMITTEE DRAFT (CD)

PROJECT NUMBER: IEC 63278-5 ED1	
DATE OF CIRCULATION: 2025-01-31	CLOSING DATE FOR COMMENTS: 2025-03-28
SUPERSEDES DOCUMENTS: 65/1032/NP, 65/1097A/RVN	

IEC TC 65 : INDUSTRIAL-PROCESS MEASUREMENT, CONTROL AND AUTOMATION	
SECRETARIAT: France	SECRETARY: Mr Didier GIARRATANO
OF INTEREST TO THE FOLLOWING COMMITTEES: SyC SM,SC 41	HORIZONTAL FUNCTION(S):
ASPECTS CONCERNED:	

This document is still under study and subject to change. It should not be used for reference purposes.

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Recipients of this document are invited to submit, with their comments, notification of any local regulations or technical reasons that may exist and should be considered should this proposal proceed, recognizing that failure to address such requirements could result in the need for new or existing "In Some Countries" clauses. (SEE [AC/22/2007](#) OR [NEW GUIDANCE DOC](#)).

TITLE: Asset Administration Shell for industrial applications – Part 5: Interfaces
--

NOTE FROM TC/SC OFFICERS:

Copyright © 2025 International Electrotechnical Commission, IEC. All rights reserved. It is permitted to download this electronic file, to make a copy and to print out the content for the sole purpose of preparing National Committee positions. You may not copy or "mirror" the file or printed version of the document, or any part of it, for any other purpose without permission in writing from IEC.

CONTENTS

1		
2		
3	FOREWORD.....	8
4	1 INTRODUCTION	10
5	1.1 General.....	10
6	1.2 Overview on parts of the series.....	10
7	2 Scope	11
8	3 Normative references	11
9	4 Terms, definitions and conventions.....	11
10	4.1 Terms and definitions.....	11
11	4.2 Conventions.....	12
12	5 Interfaces of the Asset Administration Shell.....	12
13	5.1 General.....	12
14	5.2 Requirements to the AAS interface(s) as described in IEC 63278-1 and IEC	
15	63278-4	13
16	5.3 Relationship between IEC 63278-2, IEC 63278-3 and IEC 63278-5.....	14
17	6 Breakdown of concepts to requirements of AAS interface operations.....	14
18	6.1 General.....	14
19	6.2 Provisions on interfaces.....	15
20	6.3 Provisions on identifiers.....	16
21	7 Asset Administration Shell Interfaces for AAS and Submodels.....	16
22	7.1 Overview	16
23	7.2 Naming conventions for interface operations.....	18
24	7.3 AAS administration interface and interface operations	19
25	7.3.1 General	19
26	7.3.2 Overview	20
27	7.3.3 Interface operation GetAssetAdministrationShell	20
28	7.3.4 Interface operation PutAssetAdministrationShell.....	21
29	7.3.5 Interface operation GetAllSubmodelReferences	21
30	7.3.6 Interface operation PostSubmodelReference	22
31	7.3.7 Interface operation DeleteSubmodelReference	22
32	7.3.8 Interface operation GetAssetInformation.....	23
33	7.3.9 Interface operation PutAssetInformation	23
34	7.3.10 Interface operation GetThumbnail.....	23
35	7.3.11 Interface operation PutThumbnail	24
36	7.3.12 Interface operation DeleteThumbnail	24
37	7.4 Submodel interface and interface operations	25
38	7.4.1 General	25
39	7.4.2 Overview	25
40	7.4.3 Interface operation GetSubmodel	26
41	7.4.4 Interface operation PatchSubmodel	26
42	7.4.5 Interface operation PutSubmodel.....	27
43	7.4.6 Interface Operation GetAllSubmodelElements	27
44	7.4.7 Interface operation PostSubmodelElement	28
45	7.4.8 Interface operations w.r.t. adding file content	28
46	7.4.9 Interface operation w.r.t. nested SubmodelElements	30

47	7.4.10	Interface operation w.r.t. Operation elements	34
48	7.5	AAS repository interface and interface operations	37
49	7.5.1	General	37
50	7.5.2	Overview	37
51	7.5.3	Interface operation GetAllAssetAdministrationShells	37
52	7.5.4	Interface operation GetAssetAdministrationShellById	38
53	7.5.5	Interface operation GetAllAssetAdministrationShellsByAssetId	38
54	7.5.6	Interface operation GetAllAssetAdministrationShellsByIdShort	39
55	7.5.7	Interface operation PostAssetAdministrationShell	39
56	7.5.8	Interface operation PutAssetAdministrationShellById	40
57	7.5.9	Interface operation DeleteAssetAdministrationShellById	40
58	7.6	Submodel repository interface and interface operations	41
59	7.6.1	General	41
60	7.6.2	Overview	41
61	7.6.3	Interface operation GetAllSubmodels	42
62	7.6.4	Interface operation GetSubmodelById	42
63	7.6.5	Interface operation GetAllSubmodelsBySemanticId	42
64	7.6.6	Interface operation GetAllSubmodelsByIdShort	43
65	7.6.7	Interface operation PostSubmodel	44
66	7.6.8	Interface operation PutSubmodelById	44
67	7.6.9	Interface operation PatchSubmodelById	45
68	7.6.10	Interface operation DeleteSubmodelById	45
69	8	AAS Interfaces Data types	46
70	8.1	General	46
71	8.2	Status Code, Error Handling & Result Messages	46
72	8.2.1	Generic Status Codes	46
73	8.2.2	General Result object	48
74	8.2.3	Operation objects	49
75	8.3	File Content	50
76	Annex A (informative)	Templates used in this specification	51
77	Annex B (informative)	Backus-Naur-Form	54
78	Annex C (normative)	HTTP API	56
79	C.1	General	56
80	C.2	Asset Administration Shell Interface	56
81	C.2.1	Overview	56
82	C.2.2	Interface operation GetAssetAdministrationShell	56
83	C.2.3	Interface operation PutAssetAdministrationShell	57
84	C.2.4	Interface operation GetAllSubmodelReferences	58
85	C.2.5	Interface operation PostSubmodelReference	59
86	C.2.6	Interface operation DeleteSubmodelReference	60
87	C.2.7	Interface operation GetAssetInformation	61
88	C.2.8	Interface operation PutAssetInformation	61
89	C.2.9	Interface operation GetThumbnail	62
90	C.2.10	Interface operation PutThumbnail	63
91	C.2.11	Interface operation DeleteThumbnail	64
92	C.3	Submodel Interface and Interface Operations	65
93	C.3.1	Overview	65
94	C.3.2	Interface operation GetSubmodel	66

95	C.3.3	Interface operation GetAllSubmodelElements	66
96	C.3.4	Interface operation GetSubmodelElementByPath	67
97	C.3.5	Interface operation GetFileByPath	68
98	C.3.6	Interface operation PutFileByPath	69
99	C.3.7	Interface operation DeleteFileByPath	71
100	C.3.8	Interface operation PutSubmodel	71
101	C.3.9	Interface operation PatchSubmodel	72
102	C.3.10	Interface operation PostSubmodelElement	73
103	C.3.11	Interface operation PostSubmodelElementByPath	74
104	C.3.12	Interface operation PutSubmodelElementByPath	75
105	C.3.13	Interface operation PatchSubmodelElementByPath	76
106	C.3.14	Interface operation GetSubmodelElementValueByPath	77
107	C.3.15	Interface operation DeleteSubmodelElementByPath	78
108	C.3.16	Interface operation InvokeOperationSync	79
109	C.3.17	Interface operation InvokeOperationAsync	81
110	C.3.18	Interface operation GetOperationAsyncStatus	84
111	C.3.19	Interface operation GetOperationAsyncResult	85
112	C.4	AAS repository interface and interface operations	86
113	C.4.1	Overview	86
114	C.4.2	Interface operation GetAllAssetAdministrationShells	87
115	C.4.3	Interface operation GetAssetAdministrationShellById	87
116	C.4.4	Interface operation GetAllAssetAdministrationShellsByAssetId	88
117	C.4.5	Interface operation GetAllAssetAdministrationShellsByIdShort	89
118	C.4.6	Interface operation PostAssetAdministrationShell	90
119	C.4.7	Interface operation PutAssetAdministrationShellById	91
120	C.4.8	Interface operation DeleteAssetAdministrationShellById	92
121	C.5	Submodel repository interface and interface operations	92
122	C.5.1	Overview	92
123	C.5.2	Interface operation GetAllSubmodels	93
124	C.5.3	Interface operation GetSubmodelById	94
125	C.5.4	Interface operation GetAllSubmodelsBySemanticId	95
126	C.5.5	Interface operation GetAllSubmodelsByIdShort	95
127	C.5.6	Interface operation PostSubmodel	96
128	C.5.7	Interface operation PutSubmodelById	97
129	C.5.8	Interface operation PatchSubmodelById	98
130	C.5.9	Interface operation DeleteSubmodelById	99
131	BIBLIOGRAPHY		101

132
133

FIGURES

134		
135	Figure 1 – Relationships between the documents	10
136	Figure 2 – Detailed overview of Asset Administration Shell and related entities	13
137	Figure 3 – Overview on association of different interfaces to AAS and computing	
138	environment(s).....	14
139	Figure 4 – Detailed overview of AAS and related entities regarding interfaces	15
140	Figure 5 – Services, Interfaces & APIs and Operations	17
141		
142		
143		

144

TABLES

145	Table 1 – AAS administration interface overview.....	20
146	Table 2 – Interface operation GetAssetAdministrationShell	20
147	Table 3 – Interface operation PutAssetAdministrationShell	21
148	Table 4 – Interface operation GetAllSubmodelReferences	21
149	Table 5 – Interface operation PostSubmodelReference	22
150	Table 6 – Interface operation DeleteSubmodelReference	22
151	Table 7 – Interface operation GetAssetInformation	23
152	Table 8 – Interface operation PutAssetInformation	23
153	Table 9 – Interface operation GetThumbnail	24
154	Table 10 – Interface operation PutThumbnail	24
155	Table 11 – Interface operation DeleteThumbnail	25
156	Table 12 – Submodel interface overview	25
157	Table 13 – Interface operation GetSubmodel	26
158	Table 14 – Interface operation PatchSubmodel	27
159	Table 15 – Interface operation PutSubmodel	27
160	Table 16 – Interface operation GetAllSubmodelElements	27
161	Table 17 – Interface operation PostSubmodelElement	28
162	Table 18 – Interface operation GetFileByPath	29
163	Table 19 – Interface operation PutFileByPath	29
164	Table 20 – Interface operation DeleteFileByPath	30
165	Table 21 – Interface operation GetSubmodelElementByPath	30
166	Table 22 – Interface operation PostSubmodelElementByPath	31
167	Table 23 – Interface operation PutSubmodelElementByPath	31
168	Table 24 – Interface operation PatchSubmodelElementByPath	32
169	Table 25 – Interface operation GetSubmodelElementValueByPath	33
170	Table 26 – Interface operation PatchSubmodelElementValueByPath	33
171	Table 27 – Interface operation DeleteSubmodelElementByPath	34
172	Table 28 – Interface operation InvokeOperationSync	34
173	Table 29 – Interface operation InvokeOperationAsync	35
174	Table 30 – Interface operation GetOperationAsyncStatus	36
175	Table 31 – Interface operation GetOperationAsyncResult	36
176	Table 32 – AAS repository interface overview	37
177	Table 33 – Interface operation GetAllAssetAdministrationShells	37
178	Table 34 – Interface operation GetAssetAdministrationShellById	38
179	Table 35 – Interface operation GetAllAssetAdministrationShellsByAssetId	38
180	Table 36 – Interface operation GetAllAssetAdministrationShellsByIdShort	39
181	Table 37 – Interface operation PostAssetAdministrationShell	40
182	Table 38 – Interface operation PutAssetAdministrationShellById	40
183	Table 39 – Interface operation DeleteAssetAdministrationShellById	41
184	Table 40 – Submodel repository interface overview	41
185	Table 41 – Interface operation GetAllSubmodels	42

186	Table 42 – Interface operation GetSubmodelById	42
187	Table 43 – Interface operation GetAllSubmodelsBySemanticId	43
188	Table 44 – Interface operation GetAllSubmodelsByIdShort	43
189	Table 45 – Interface operation PostSubmodel.....	44
190	Table 46 – Interface operation PutSubmodelById.....	44
191	Table 47 – Interface operation PatchSubmodelById	45
192	Table 48 – Interface operation DeleteSubmodelById	45
193	Table 49 – Generic status Codes	46
194	Table 50 – General Result object	48
195	Table 51 – Message object	48
196	Table 52 – Enumeration MessageType	48
197	Table 53 – OperationResult object	49
198	Table 54 – Enumeration ExecutionState	49
199	Table 55 – OperationHandle object.....	50
200	Table A.1 – Interface Description	51
201	Table A.2 – Interface operation Description	51
202	Table A.3 – Data Types for Payload Description	52
203	Table A.4 – Enumeration Description	53
204	Table C.1 – AAS interface operations	56
205	Table C.2 – AAS Submodel interface operations	65
206	Table C.1 – AAS repository interface operations	86
207	Table C.1 – Submodel repository interface operations	93
208		
209		
210		

INTERNATIONAL ELECTROTECHNICAL COMMISSION

ASSET ADMINISTRATION SHELL FOR INDUSTRIAL APPLICATIONS

Part 5: Interfaces

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 not 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> shall not be held responsible for identifying any or all such patent rights.

IEC 63278-5 Ed.1 has been prepared by 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
65/XX/FDIS	65/XX/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.

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

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

- 269 • reconfirmed,
- 270 • withdrawn,
- 271 • replaced by a revised edition, or
- 272 • amended.
- 273 •
- 274

1 INTRODUCTION

1.1 General

This document is the fifth part of the series "Asset Administration Shell for Industrial Applications". The Asset Administration Shell (AAS) is seen as one interoperable manifestation of a digital twin in manufacturing.

1.2 Overview on parts of the series

The IEC 63278 series Asset Administration Shell for Industrial Applications includes the following parts covering the following topics:

- Part 1: Asset Administration Shell structure (IEC 63278-1 ED1)
- Part 2: Information meta model (IEC 63278-2)
- Part 3: Security provisions for Asset Administration Shells (IEC 63278-3)
- Part 4: Use Cases and Modelling Examples (IEC 63278-4)
- Part 5: Interfaces (IEC 63278-5, this document)

Figure 1 illustrates the relationships between the documents currently being worked on.

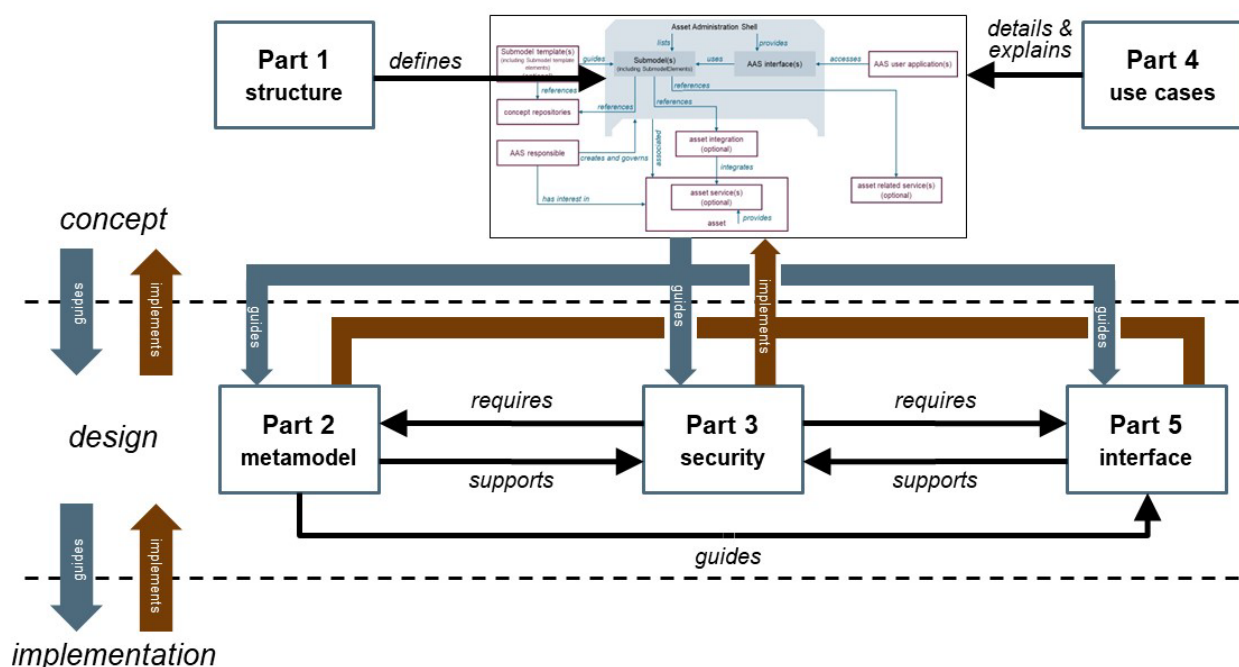


Figure 1 – Relationships between the documents

- Part 1 and Part 4 address the concept. Part 1 describes the conceptual framework for the AAS, the structure of the AAS and requirements associated to the AAS and related entities. Part 4 details Part 1 by describing more detailed use cases and illustrates the concept of the AAS by selected modeling examples.
- Part 2, Part 3, and Part 5 address the design:

- Part 2 formalizes the requirements of Part 1 for Submodels and Submodel templates to a clearly defined information metamodel. Furthermore, Part 2 considers the specific requirements provided by Part 3.

- Part 3 addresses the cross-cutting concern of security. Part 3 formalizes the requirements of Part 1 and use cases of Part 4 and considers the modeling concepts provided by Part 2 and functions provided by this Part “interface”.

NOTE Further cross-cutting concerns could be considered in the future, such as uniform version management.

- Part 5 will formalize the concepts and the requirements of Part 1 for the AAS to a clearly defined set of functions of the interface. Part 5 is also subject to guidance of Part 2 and considers the specific requirements provided by Part 3.

2 Scope

IEC 63278 series is standardizing a digital representation of an asset, called Asset Administration Shell (AAS). This part 5 of the IEC 63278 series specifies interfaces of the Asset Administration Shell (AAS).

3 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 63278-1 ED1 Asset Administration Shell for industrial applications – Part 1: Asset Administration Shell structure

IEC 63278-2 ED1 Asset Administration Shell for Industrial Applications – Part 2: Information meta model

IEC 63278-3 ED1 Asset Administration Shell for Industrial Applications – Part 3: Security provisions for Asset Administration Shells

IEC 63278-4 ED1 Asset Administration Shell for Industrial Applications Part 4: Use Cases and Modelling Examples

4 Terms, definitions and conventions

4.1 Terms and definitions

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

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

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

For the purposes of this document, the terms and definitions given in IEC TR 63283-1:2022 [6] Industrial-process measurement, control and automation – Smart manufacturing - Part 1: Terms and definitions and IEC 63278-1:2024 apply.

4.1.1

AAS administration interface

AAS interface allowing to access one or multiple Asset Administration Shells

4.1.2

AAS repository

set of multiple AAS within a computing environment

4.1.3

AAS Submodel interface

AAS interface allowing to access Submodels of an Asset Administration Shell

4.1.4

computing environment

software service providing interfaces to create, register/unregister and lookup Asset Administration Shells

4.1.5

interface

unit of reusability across services

4.1.6

interface operation

interaction pattern defined with respect to a specific interface

4.1.7

service

distinct part of the functionality that is provided by an entity through interfaces

[SOURCE: IEC 60050-741:2020, 741-01-28]

4.1.8

Submodel repository

set of multiple Submodels within a computing environment

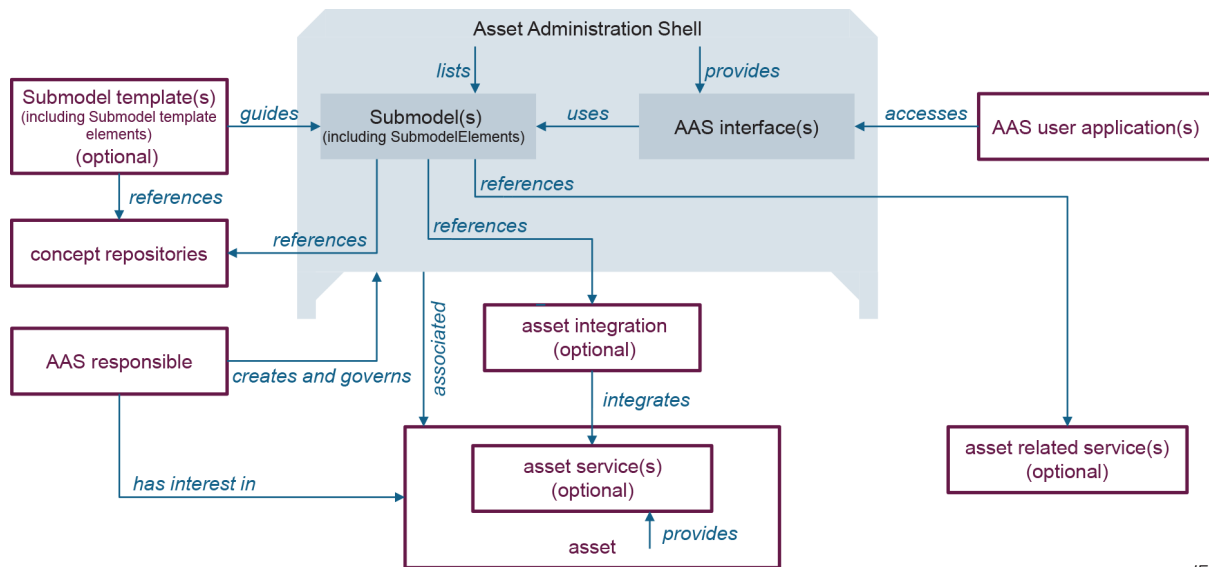
4.2 Conventions

For some terms, such as Asset Administration Shell or SubmodelElement, start case or pascal case is used, in order to distinguish them from English text.

5 Interfaces of the Asset Administration Shell

5.1 General

IEC 63278-1 defines the overall structure of the AAS and related entities as shown in Figure 2.



IEC

Figure 2 – Detailed overview of Asset Administration Shell and related entities

Part 5 uses Part 1 and Part 4 as a basis to design the AAS interface(s) and its operations.

NOTE The Asset Administration Shell lists Submodel(s) by referencing to Submodel identifiers.

5.2 Requirements to the AAS interface(s) as described in IEC 63278-1 and IEC 63278-4

The purpose of this clause is to list the requirements stated and explained in Part 1 and Part 4 and thereby setting the focus for Part 5.

An AAS interface makes the information and services of the Asset Administration Shell available externally. The following requirements to the AAS interface are described in IEC 63278-1:

- All Asset Administration Shells shall provide a set of standardised AAS interfaces to enable AAS user applications to create, delete and update an Asset Administration Shell and the Submodels and SubmodelElements in a uniform way.
- All Asset Administration Shells shall provide a set of standardised AAS interfaces to enable AAS user applications to retrieve an Asset Administration Shell not only by the unique identifier of the asset it is representing but also by alternative identifiers of the same asset. These alternative identifiers shall be unique in a given namespace only, but not globally.
- All Asset Administration Shells shall provide a set of standardised AAS interfaces to enable AAS user applications to access the information and to access and execute the services they are offering in a uniform way.

These requirements are detailed in IEC 63278-4, clause 4.7, with respect to data exposure and accountability.

NOTE 1 The term “AAS interface(s)” in IEC 63278-1 and IEC 63278-4 refers in the context of this document to interface operations (confer 7.1), which are defined with respect to different interfaces.

NOTE 2 The notion of “create, delete and update” of elements applies depending on further conditions, such as permissions and access rights.

NOTE 3 The availability of such alternative identifiers is not always given. See IEC 63278-2 for “specificAssetId”.

5.3 Relationship between IEC 63278-2, IEC 63278-3 and IEC 63278-5

The terms and definitions, meta model entities and data types for the parameters of the interface operations and return types defined in clause 6 are defined in Part 2.

In later stages of the process, statements regarding relationships between Part 3 and Part 5 might be added.

6 Breakdown of concepts to requirements of AAS interface operations

6.1 General

In order to provide interfaces to be accessed by AAS user applications, a technical implementation is realized in a computing environment. The computing environment varies and depends on the decisions taken by the AAS responsible(s) and other persons in the operating organisation (see Figure 3).

The entirety of these realized interface operations (confer section 7) allows AAS user applications to (among others) create, update, retrieve, delete Asset Administration Shells and Submodels and dependent entities (CRUD principle, purpose of information, relationship). Also, the invocation of Operation elements is allowed (purpose service, see IEC 63278-2).

In later stages of the process, further computing environment interfaces might be added to the scope.

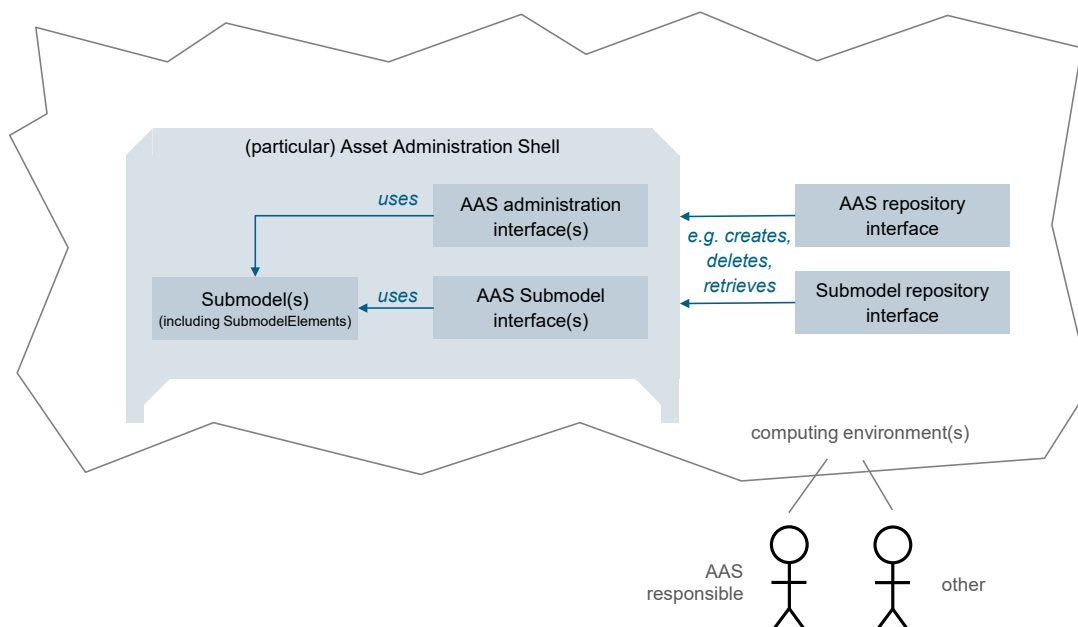


Figure 3 – Overview on association of different interfaces to AAS and computing environment(s)

Figure 4 extends Figure 2 with additional entities regarding the specification of the interfaces of the AAS:

- The AAS interface logically consists of AAS administration interface(s) and AAS Submodel interface(s). The AAS administration interface(s) allow accessing one or multiple AASs (see 7.3). The AAS Submodel interface(s) allow accessing (typically) multiple Submodels (see 7.4).
- From a design perspective, the computing environment is a software service, which allows creating and accessing one or multiple Asset Administration Shells. The interface(s) of this software service are called computing environment interface(s) and allow, for example, to create, update and retrieve one or multiple AAS or further elements.
- The set of multiple AAS within a computing environment is designated as AAS repository. The set of multiple Submodels within a computing environment is designated as Submodel repository.
- For this document, two computing environment interfaces are specified: AAS repository interface and Submodel repository interface. The AAS repository interface allows to create, update and retrieve one or multiple AAS (see 7.5), while the Submodel repository interface allows to create, update and retrieve one or multiple Submodels (see 7.6).

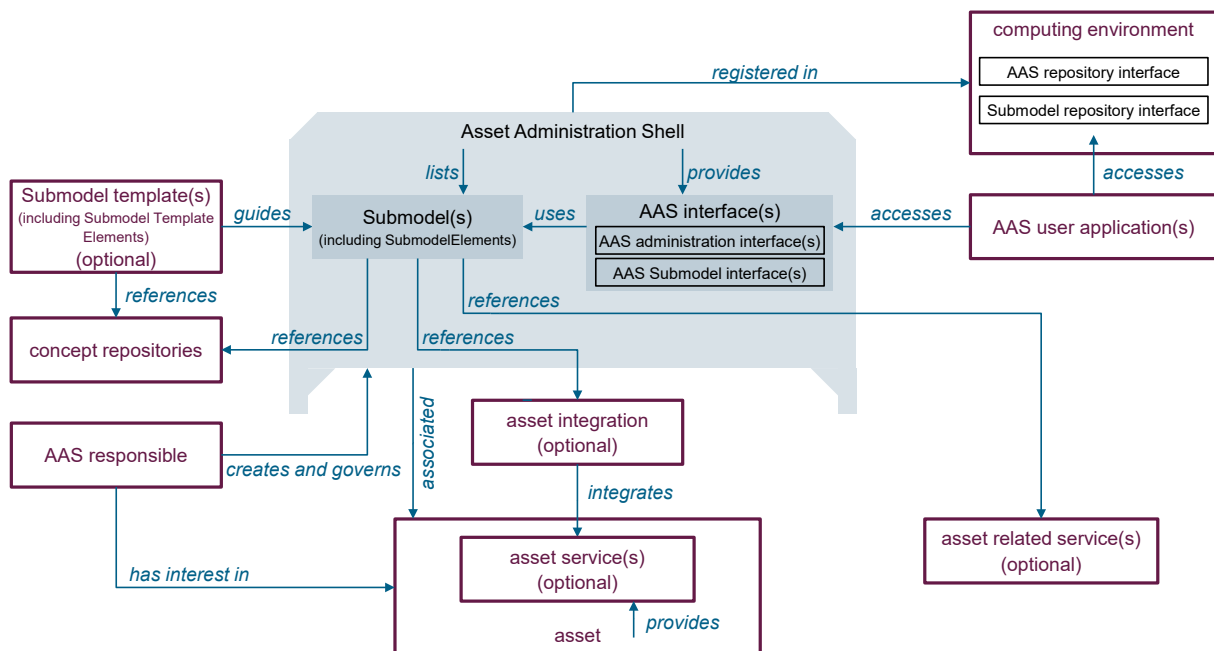


Figure 4 – Detailed overview of AAS and related entities regarding interfaces

6.2 Provisions on interfaces

The above-described mechanism of access requires the AAS user application to specifically know the identifiers of AAS and/ or Submodels beforehand.

The above interfaces allow accessing one or multiple entities. It is up to the computing environment or the applied use cases to restrict this access.

EXAMPLE It is possible, that the computing environment within an embedded device restricts the access to exactly one AAS for the specific device.

EXAMPLE It is possible, that the computing environment within an embedded device restricts the access to read operations, as the storage of the device does not allow writing information.

NOTE Restriction of access is realized by returning an error information by the specific interface operation.

449

450 6.3 Provisions on identifiers

451 In order to identify Asset Administration Shells and Submodels, a specific identifier is required
452 which shall be globally distinct (see IEC 63278-1, section 5).

453 These identifiers are required for access, but also for creation of the above entities. For the
454 creation of above entities, these identifiers need to be known or created beforehand, e.g. by
455 the AAS responsible or the organization providing the computing environment.

456 NOTE 1 In IEC 63278-1, section 5.2.3, the creation of these identifiers is described for an organisation.

457 NOTE 2 The retrieval by global asset identifier (see IEC 63278-2) is not part of this document.

458

459 7 Asset Administration Shell Interfaces for AAS and Submodels

460 7.1 Overview

461 This document uses the Service Model as illustrated in Figure 5 for a uniform understanding
462 and naming. It basically distinguishes between associated concepts on several levels (from left
463 to right):

- 464 • technology-neutral level: concepts that are independent from selected technologies;
- 465 • technology-specific level: concepts that are instantiated for a given technology and/or
466 architectural style (e.g. HTTP/REST, OPC UA, MQTT);
- 467 • implementation level: concepts that are related to an implementation architecture that
468 comprises one or more technologies (e. g. C#, C++, Java, Python);
- 469 • runtime level: concepts that are related to identifiable components in an operational
470 smart manufacturing system.

471 This section deals with the concepts of the **technology-neutral level**. Annex C adds
472 technology-specific details for HTTP/ REST in order to provide interoperability. However, to
473 avoid terminological and conceptual misunderstandings, a whole service model is provided here
474 (see Figure 5).

475

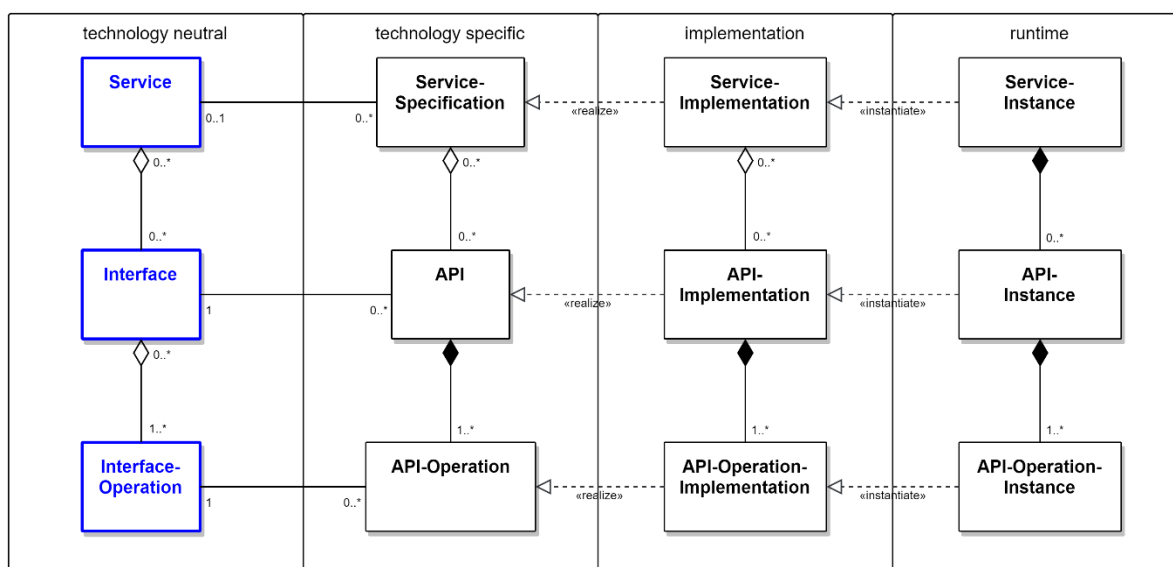


Figure 5 – Services, Interfaces & APIs and Operations

The technology-neutral level in Figure 5 comprises the following concepts:

- **Service:** a service describes a demarcated scope of functionality (including its informational and non-functional aspects), which is offered by an entity or organization via interfaces;
- **Interface:** this is the most important concept as it is understood to be the unit of reusability across services and the unit of standardization when mapped to application programming interfaces (API) in the technology-specific level. One interface may be mapped to several APIs depending on the technology and architectural style used, e.g. HTTP/REST or OPC UA, whereby these API mappings also need to be standardized for the sake of interoperability;
- **Interface-Operation:** interface operations define interaction patterns via the specified interface.

The technology-specific level comprises the following concepts:

- **Service Specification:** specification of a service according to the notation, architectural style, and constraints of a selected technology. Among others, it comprises and refers to the list of APIs that forms this service specification. These may be interoperable standard APIs but also other, proprietary APIs;

NOTE Such a technology-specific service specification may be but does not have to be derived from the “service” described in the technology-neutral form. It is up to the system architect and service engineer to tailor the technology-specific service according to the needs of the use cases.

- **API:** specification of the set of operations and events that forms an API in a selected technology. It is derived from the interface description on the technology-neutral level. Hence, if there are several selected technologies, one interface may be mapped to several APIs;
- **API-Operation:** specification of the operations (procedures) that may be called through an API. It is derived from the interface operation description on the technology-neutral level. When selecting technologies, one interface operation may be mapped to several API-operations; several interface operations may also be mapped to the same API-operation.

The implementation level comprises the following concepts:

- **Service-Implementation:** service realized in a selected implementation language following the specification in the Service Specification description on the technology-specific level;
- **API-Implementation:** set of operations realized in a selected implementation language following the specification in the API description on the technology-specific level;
- **API-Operation-Implementation:** concrete realization of an operation in a selected implementation language following the specification in the API-Operation description on the technology-specific level.

The runtime level comprises the following concepts:

- **Service-Instance:** instance of a Service-Implementation including its API-Instances for communication. Additionally, it has an identifier to be identifiable within a given context;
- **API-Instance:** instance of an API-Implementation which has an endpoint to get the information about this instance and the related operations;
- **API-Operation-Instance:** instance of an API-Operation-Implementation which has an endpoint to get invoked.

7.2 Naming conventions for interface operations

The interface operations follow a resource-oriented approach. Resources are the elements of Asset Administration Shells described in IEC 63278-2. The approach consists of three main agreements:

- The interface is stateless: Each interface operation is independent. If required, state data will be handed back to the AAS user application to be managed by the client and handed over again for the next interface operation.
- Resources (nouns): each resource is a clearly defined noun. This means that it has a specific name and its relation to other nouns is defined. The nouns and the relationships between them are taken from the list of referable objects of IEC 63278-2.
- Operation-Verbs: a small set of Operation-Verbs (Get, GetAll, Post, Put, Patch, Delete, Invoke).

The **Operation Verbs** are:

- **Get:** A Get returns a single resource based on the resource identifier which is the identifier for identifiables and the path of referables.
- **GetAll:** Returns a list of resources based on optionally available parameters such as filters.
- **Post:** Creates a new resource. The identifier of the resource is part of the resource description.
- **Put:** Replaces an existing resource.

- Patch: Updates an existing resource.
- Delete: Deletes a resource based on a given identifier.
- Invoke: Invokes an operation at a specified path.

NOTE In some interface operations, the term “payload” designates the resource, that is, the AAS element or file content, which is to be stored or retrieved. The payload therefore identifies, what is to be transported.

Naming rules for interface operations:

The following rules shall apply for the operation names described in section 7 and following (see Annex B for Backus-Naur-Form):

```
<Interface Operation> ::= <Operation Verb><Model Element  
                        Name>{[<Modifier>]}*  
                        ["By"<By-Qualifier>]  
  
<Operation Verb>      ::= "Get" | "GetAll" | "Put" | "Post" | "Patch" | "Delete" |  
                        "Invoke"  
  
<Model Element Name>  ::= "AssetAdministrationShell" | "SubmodelReference"  
                        ["s"] |  
                        "AssetInformation" | "Submodel" |  
                        "SubmodelElement"["s"] | "Operation" | "Thumbnail"  
  
<Modifier>            ::= "Value" | "Sync" | "Async" | "Status" | "Result"  
  
<By-Qualifier>        ::= "Path"
```

Example 1 GetSubmodel has operation verb “Get” and model element name “Submodel”.

Example 2 GetAllSubmodelElementsByPath has operation verb “GetAll” and model element name “SubmodelElements” plus a by-qualifier “Path”.

Some interface operations are expecting a **IdShortPath** input (see 7.4.9.1). The **IdShortPath** is a list of IdShort identifiers, that represents the hierarchy path to a specific SubmodelElement within a Submodel.

7.3 AAS administration interface and interface operations

7.3.1 General

These interface operations make it possible to access the elements of Asset Administration Shells, which are each an Identifiable entity. The format of the tables in this section are described in Annex A.

NOTE For Identifiable, see IEC 63278-2.

7.3.2 Overview

Table 1 provides an overview of the interface operations of the Asset Administration Shell, which are then explained in the following subsections including details such as expected input and output parameters.

Table 1 – AAS administration interface overview

Interface operation name	Description
GetAssetAdministrationShell	Returns the Asset Administration Shell
PutAssetAdministrationShell	Replaces the current Asset Administration Shell
GetAllSubmodelReferences	Returns all Submodel References
PostSubmodelReference	Creates a Submodel Reference at the Asset Administration Shell
DeleteSubmodelReference	Deletes a specific Submodel Reference from the Asset Administration Shell
GetAssetInformation	Returns the Asset Information
PutAssetInformation	Replaces the Asset Information
GetThumbnail	Returns the thumbnail file
PutThumbnail	Replaces the thumbnail file
DeleteThumbnail	Deletes the thumbnail

7.3.3 Interface operation GetAssetAdministrationShell

The interface operation described in Table 2 allows to request for a specific AAS based on the provided AAS identifier.

Table 2 – Interface operation GetAssetAdministrationShell

Interface Operation Name	GetAssetAdministrationShell			
Explanation	Returns the Asset Administration Shell associated with the provided AAS identifier			
Name	Description	Mand.	Type	Card.
Input Parameter				
aasId	AssetAdministrationShell identifier	yes	AssetAdministrationShellID	1
Output Parameter				
statusCode	Status code	yes	StatusCode	1
payload	Requested Asset Administration Shell	yes	AssetAdministrationShell	1

7.3.4 Interface operation PutAssetAdministrationShell

The interface operation described in Table 3 allows to replace/update a complete specific AAS.

Table 3 – Interface operation PutAssetAdministrationShell

Interface Operation Name	PutAssetAdministrationShell			
Explanation	Replaces an specific Asset Administration Shell that is associated with the provided AAS identifier			
Name	Description	Mand.	Type	Card.
Input Parameter				
aasId	AssetAdministrationShell identifier	yes	AssetAdministrationShellID	1
aas	AssetAdministrationShell	yes	Provided new Asset Administration Shell that should be used instead	1
Output Parameter				
statusCode	Status Code	yes	StatusCode	1
payload	AssetAdministrationShell	yes	Replaced Asset Administration Shell	1

7.3.5 Interface operation GetAllSubmodelReferences

The interface operation described in Table 4 allows to return all Submodel references contained in a specific AAS.

Table 4 – Interface operation GetAllSubmodelReferences

Interface Operation Name	GetAllSubmodelReferences			
Explanation	Returns all Submodel References of an associated AAS			
Name	Description	Mand.	Type	Card.
Input Parameter				
aasId	AssetAdministrationShell identifier	yes	AssetAdministrationShellID	1
Output Parameter				
statusCode	Status code	yes	StatusCode	1
payload	Requested Submodel References	yes	Reference	0..*

7.3.6 Interface operation PostSubmodelReference

The interface operation described in Table 5 enhances the list of Submodel references of an associated AAS by a new provided Submodel reference.

Table 5 – Interface operation PostSubmodelReference

Interface Operation Name	PostSubmodelReference			
Explanation	Creates a Submodel Reference at the Asset Administration Shell			
Name	Description	Mand.	Type	Card.
Input Parameter				
aasId	AssetAdministrationShell identifier	yes	AssetAdministrationShellID	1
submodelRef	Reference to the Submodel	yes	Reference	1
Output Parameter				
statusCode	Status code	yes	StatusCode	1
payload	Created Submodel Reference	yes	Reference	1

7.3.7 Interface operation DeleteSubmodelReference

The interface operation described in Table 6 removes a specific Submodel reference from the list of Submodel references of an associated AAS.

Table 6 – Interface operation DeleteSubmodelReference

Interface Operation Name	DeleteSubmodelReference			
Explanation	Deletes the Submodel Reference from the Asset Administration Shell			
Name	Description	Mand.	Type	Card.
Input Parameter				
aasId	AssetAdministrationShell identifier	yes	AssetAdministrationShellID	1
submodelId	The unique id of the Submodel for the reference to be deleted	yes	Identifier	1
Output Parameter				
statusCode	Status code	yes	StatusCode	1

7.3.8 Interface operation GetAssetInformation

The interface operation described in Table 7 returns related asset information of an associated AAS.

Table 7 – Interface operation GetAssetInformation

Interface Operation Name	GetAssetInformation			
Explanation	Returns the Asset Information			
Name	Description	Mand.	Type	Card.
Input Parameter				
aasId	AssetAdministrationShell identifier	yes	AssetAdministrationShellID	1
Output Parameter				
statusCode	Status code	yes	StatusCode	1
payload	Requested Asset Information	yes	AssetInformation	1

7.3.9 Interface operation PutAssetInformation

The interface operation described in Table 8 allows to set or update asset-based information of an associated AAS.

Table 8 – Interface operation PutAssetInformation

Interface Operation Name	PutAssetInformation			
Explanation	Replaces/updates Asset Information			
Name	Description	Mand.	Type	Card.
Input Parameter				
aasId	AssetAdministrationShell identifier	yes	AssetAdministrationShellID	1
assetInfo	Asset Information	yes	AssetInformation	1
Output Parameter				
statusCode	Status code	yes	StatusCode	1

7.3.10 Interface operation GetThumbnail

The interface operation described in Table 9 allows to request for a thumbnail file of an associated AAS.

Table 9 – Interface operation GetThumbnail

Interface Operation Name	GetThumbnail			
Explanation	Returns the thumbnail file			
Name	Description	Mand.	Type	Card.
Input Parameter				
aasId	AssetAdministrationShell identifier	yes	AssetAdministrationShellID	1
Output Parameter				
statusCode	Status code	yes	StatusCode	1
payload	Requested thumbnail file	yes	File Content	1

NOTE Thumbnail refers to a reduced size image of the asset (see IEC 63278-2).

7.3.11 Interface operation PutThumbnail

The interface operation described in Table 10 allows to set or update a thumbnail file of an associated AAS.

Table 10 – Interface operation PutThumbnail

Interface Operation Name	PutThumbnail			
Explanation	Replaces the thumbnail file			
Name	Description	Mand.	Type	Card.
Input Parameter				
aasId	AssetAdministrationShell identifier	yes	AssetAdministrationShellID	1
file	Thumbnail file	yes	File Content	1
Output Parameter				
statusCode	Status code	yes	StatusCode	1

7.3.12 Interface operation DeleteThumbnail

The interface operation described in Table 11 allows to remove a thumbnail file of an associated AAS.

Table 11 – Interface operation DeleteThumbnail

Interface Operation Name	DeleteThumbnail			
Explanation	Deletes the thumbnail file			
Name	Description	Mand.	Type	Card.
Input Parameter				
aasId	AssetAdministrationShell identifier	yes	AssetAdministrationShellID	1
Output Parameter				
statusCode	Status code	yes	StatusCode	1

7.4 Submodel interface and interface operations

7.4.1 General

These interface operations make it possible to access the elements of Submodels, which are each an Identifiable entity. The format of the tables in this section are described in Annex A.

7.4.2 Overview

Table 12 provides an overview of the interface operations of the Submodel, which are then explained in the following subsections including details such as expected input and output parameters.

Table 12 – Submodel interface overview

Interface Operation Name	Description
GetSubmodel	Returns the Submodel
GetAllSubmodelElements	Returns all SubmodelElements including their hierarchy
GetSubmodelElementByPath	Returns a specific SubmodelElement from the Submodel at a specified path
GetFileByPath	Returns a specific file from the Submodel at a specified path
PutFileByPath	Replaces the file of an existing SubmodelElement at a specified path within the SubmodelElement hierarchy
DeleteFileByPath	Deletes the file of an existing SubmodelElement at a specified path within the SubmodelElement hierarchy
PutSubmodel	Replaces the Submodel
PatchSubmodel	Updates the Submodel
PostSubmodelElement	Creates a new SubmodelElement as a child of the Submodel. The idShort of the new SubmodelElement shall be set in the payload.
PostSubmodelElementByPath	Creates a new SubmodelElement at a specified path within the SubmodelElements hierarchy. The idShort of the new SubmodelElement shall be set in the payload.

Interface Operation Name	Description
PutSubmodelElementByPath	Replaces an existing SubmodelElement at a specified path within the SubmodelElement hierarchy
PatchSubmodelElementByPath	Updates an existing SubmodelElement at a specified path within the SubmodelElement hierarchy
GetSubmodelElementValueByPath	Returns the value of the SubmodelElement at a specified path according to the protocol-specific RAW-value payload
DeleteSubmodelElementByPath	Deletes a SubmodelElement at a specified path within SubmodelElement hierarchy
InvokeOperationSync	Synchronously invokes an Operation at a specified path with a client timeout in ms
InvokeOperationAsync	Asynchronously invokes an Operation at a specified path with a client timeout in ms
GetOperationAsyncStatus	Returns the current status of an asynchronously invoked operation
GetOperationAsyncResult	Returns the OperationResult of an asynchronously invoked operation

635

636 **7.4.3 Interface operation GetSubmodel**

637 The interface operation described in Table 13 allows to request for a specific Submodel based
 638 on the provided Submodel identifier.

639 **Table 13 – Interface operation GetSubmodel**

Interface Operation Name	GetSubmodel			
Explanation	Returns the Submodel			
Name	Description	Mand.	Type	Card.
Input Parameter				
submodelId	Submodel identifier	yes	SubmodelID	1
Output Parameter				
statusCode	Status code	yes	StatusCode	1
payload	Requested Submodel	yes	Submodel	1

640

641 **7.4.4 Interface operation PatchSubmodel**

642 Th interface operation described in Table 14 allows to update a specific Submodel identified by
 643 a Submodel identifier.

Table 14 – Interface operation PatchSubmodel

Interface Operation Name	PatchSubmodel			
Explanation	Updates the Submodel			
Name	Description	Mand.	Type	Card.
Input Parameter				
submodelId	Submodel identifier	yes	SubmodelID	1
submodel	Updated Submodel object	yes	Submodel	1
Output Parameter				
statusCode	Status code	yes	StatusCode	1
payload	Updated Submodel	yes	Submodel	1

7.4.5 Interface operation PutSubmodel

The interface operation described in Table 15 allows to replace a specific Submodel identified by a Submodel identifier.

Table 15 – Interface operation PutSubmodel

Interface Operation Name	PutSubmodel			
Explanation	Replaces the Submodel			
Name	Description	Mand.	Type	Card.
Input Parameter				
submodelId	Submodel identifier	yes	SubmodelID	1
Output Parameter				
statusCode	Status code	yes	StatusCode	1
payload	Replaced Submodel	yes	Submodel	1

7.4.6 Interface Operation GetAllSubmodelElements

The interface operation described in Table 16 allows to request for a specific Submodel and its (hierarchical) elements based on a provided Submodel identifier.

Table 16 – Interface operation GetAllSubmodelElements

Interface Operation Name	GetAllSubmodelElements			
Explanation	Returns all SubmodelElements including their hierarchy			

Name	Description	Mand.	Type	Card.
Input Parameter				
submodelId	Submodel identifier	yes	SubmodelID	1
Output Parameter				
statusCode	Status code	yes	StatusCode	1
payload	Requested SubmodelElements	yes	SubmodelElement	0..*

655

656 **7.4.7 Interface operation PostSubmodelElement**

657 The interface operation described in Table 17 allows to enhance a specific Submodel with a
 658 new SubmodelElement at the root level of the Submodel.

659 **Table 17 – Interface operation PostSubmodelElement**

Interface Operation Name	PostSubmodelElement			
Explanation	Creates a new SubmodelElement as a child of the Submodel. The idShort of the new SubmodelElement shall be set in the payload.			
Name	Description	Mand.	Type	Card.
Input Parameter				
submodelId	Submodel identifier	yes	SubmodelID	1
submodelElement	Submodel element	yes	SubmodelElement	1
Output Parameter				
statusCode	Status code	yes	StatusCode	1
payload	Created SubmodelElement	yes	SubmodelElement	1

660

661 **7.4.8 Interface operations w.r.t. adding file content**662 **7.4.8.1 General**

663 SubmodelElements of subclass File allow adding information, which is represented by separate
 664 file content and is of purpose information (see IEC 63278-2). AAS user applications access
 665 these contents.

666 **EXAMPLE** A handbook of a machine in PDF format is stored and retrieved as file content by AAS user application
 667 such as maintenance software.

668 The interface operations in this subsection concern about the storage and retrieval of such file
 669 content with respect to File elements within a Submodel.

7.4.8.2 Interface operation GetFileByPath

The interface operation described in Table 18 allows to request for a specific AAS user related file representation as a specific SubmodelElement, which is identified by a provided Submodel identifier and IdShortPath.

Table 18 – Interface operation GetFileByPath

Interface Operation Name	GetFileByPath			
Explanation	Returns a specific file from the Submodel at a specified path			
Name	Description	Mand.	Type	Card.
Input Parameter				
submodelId	Submodel identifier	yes	SubmodelID	1
Path	Path to a SubmodelElement	yes	Key	1..*
Output Parameter				
statusCode	Status code	yes	StatusCode	1
payload	Requested file	yes	File Content	0..1

7.4.8.3 Interface operation PutFileByPath

The interface operation described in Table 19 allows to set or update a specific AAS user related file at a specific SubmodelElement, which is identified by a provided Submodel identifier and IdShortPath and that expects a file representation.

Table 19 – Interface operation PutFileByPath

Interface Operation Name	PutFileByPath			
Explanation	Replaces the file of an existing SubmodelElement at a specified path within the SubmodelElement hierarchy			
Name	Description	Mand.	Type	Card.
Input Parameter				
submodelId	Submodel identifier	yes	SubmodelID	1
path	Path to a SubmodelElement	yes	Key	1..*
payload	Replacing file	yes	File Content	1
Output Parameter				
statusCode	Status code	yes	StatusCode	1