



DPC: 25 / 30509284 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: 21 February 2025

Origin: International

Latest date for receipt of comments: 12 March 2025

Project No. 2025/00307

Responsible committee: EPL/80 Maritime navigation and radiocommunication equipment and systems

Interested committees:

Title: Draft BS EN 63173-2 ed. 2.0 Maritime navigation and radiocommunication equipment and systems - Data interfaces

Part 2: Secure communication between ship and shore (SECOM)

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 12 March 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: **Mr Simon Bounds (BSI)**
Direct tel: **0208 996 7281**
E-mail: simon.bounds@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.	



80/1149/CD

COMMITTEE DRAFT (CD)

PROJECT NUMBER: IEC 63173-2 ED2	
DATE OF CIRCULATION: 2025-02-14	CLOSING DATE FOR COMMENTS: 2025-04-11
SUPERSEDES DOCUMENTS: 80/1145/RR	

IEC TC 80 : MARITIME NAVIGATION AND RADIOCOMMUNICATION EQUIPMENT AND SYSTEMS	
SECRETARIAT: United Kingdom	SECRETARY: Mr Kim Fisher
OF INTEREST TO THE FOLLOWING COMMITTEES:	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:

Maritime navigation and radiocommunication equipment and systems - Data interfaces - Part 2: Secure communication between ship and shore (SECOM)

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

FOREWORD	13
INTRODUCTION	15
1 Scope	17
2 Normative references	17
3 Terms, definitions and abbreviated terms	17
3.1 Terms and definitions	18
3.2 Generic name of digital signature algorithm in this document	19
3.3 Abbreviated terms	21
4 General description of SECOM.....	22
4.1 General	22
4.2 Information service interface.....	23
4.3 Information security	23
4.3.1 Measures	23
4.3.2 SECOM PKI	24
4.3.3 Communication channel security	24
4.3.4 Data protection.....	24
4.3.5 Certificate revocation status	26
4.4 Service discoverability	26
4.5 Structure of this document	27
5 SECOM information service interface	27
5.1 General	27
5.2 How to read descriptions of service interface definition	28
5.3 Service technology and service transportation protocol	29
5.4 Service interface versioning	29
5.5 Pagination	30
5.6 Common information objects and data types.....	30
5.6.1 General	30
5.6.2 Basic data types.....	30
5.6.3 SECOM_ExchangeMetadataObject	31
5.6.4 Transfer of public key	32
5.6.5 PaginationObject.....	34
5.6.6 ContainerTypeEnum.....	35
5.6.7 SECOM_DataProductType	35
5.6.8 SECOM_ResponseCodeEnum	36
5.6.9 AckRequest Enum	36
5.6.10 Common HTTP response codes	36
5.6.11 Well-known text - WKT	36
5.6.12 Universally Unique Identifier - UUID	37
5.6.13 UN/LOCODE	38
5.7 Service interface definitions	39
5.7.1 General	39
5.7.2 Service interface - Upload	39
5.7.3 Service interface - Upload Link.....	45
5.7.4 Service interface - Acknowledgement.....	50

5.7.5	Service interface - Get	54
5.7.6	Service interface - Get Summary	59
5.7.7	Service interface - Get By Link	62
5.7.8	Service interface - Access	64
5.7.9	Service interface - Access Notification	67
5.7.10	Service interface - Subscription	69
5.7.11	Service interface - Remove Subscription	73
5.7.12	Service interface - Subscription Notification	75
5.7.13	Service interface - Capability	78
5.7.14	Service interface - Ping	81
5.7.15	Service interface - EncryptionKey	83
5.7.16	Service interface - PublicKey	89
6	SECOM communication channel security	92
6.1	General	92
6.2	Secure transfer	93
6.2.1	Secure communication channel	93
6.2.2	Authentication procedure	93
7	SECOM data protection	94
7.1	General	94
7.2	Data compression and packaging	94
7.3	Data authentication and signing	95
7.3.1	General	95
7.3.2	Data formats and standards for digital signatures, keys and certificates 95	
7.3.3	Creation of digital signature	96
7.3.4	Creation of envelope signature	96
7.3.5	Verification of digital signature	98
7.3.6	Verification of envelope signature	98
7.3.7	Example of commands for data authentication	99
7.4	Data encryption	99
7.4.1	General	99
7.4.2	Encryption algorithm	99
7.5	Creation and transfer of encryption key	99
7.5.1	General	99
7.5.2	SECOM encryption key management	100
7.5.3	Generate encryption key	102
7.5.4	Sign the protected encryption key	102
7.5.5	Transfer of the encryption key	102
7.5.6	Example	102
8	SECOM PKI	103
8.1	General	103
8.2	Scheme	103
8.2.1	General	103
8.2.2	Scheme administrator	103
8.2.3	Data servers	103
8.2.4	Data clients	104
8.2.5	Procedure	104
8.3	Generation of public and private key	104

8.4	Certificate signing request	105
8.5	Certificate revocation	105
8.5.1	General	105
8.5.2	CRL - Certificate revocation list	105
8.5.3	OCSP - Online certificate status protocol	105
8.6	SECOM PKI service interface	105
8.6.1	General	105
8.6.2	Service interface - CSR	106
8.6.3	Service interface - GetPublicKey	108
8.6.4	Service interface - CRL	110
8.6.5	Service interface - OCSP	111
8.6.6	Service interface - Revoke	113
9	SECOM service discovery service interface	115
9.1	General	115
9.2	Service interface - Search service	115
9.2.1	Specification	115
9.2.2	Data exchange model	116
9.2.3	REST design	118
10	SECOM error cases	119
10.1	Error cases	119
10.2	General	119
10.3	Message integrity	119
10.4	Data integrity	120
10.5	Transport confidentiality	120
10.6	Data protection	120
10.7	Service identity	121
10.8	Client identity	121
10.9	Client authorization	121
10.10	Bandwidth optimization	121
10.11	Large message transfer	121
10.12	Closed loop communication	122
10.13	Service discoverability	123
10.14	Information push	123
10.15	Information pull	123
10.16	Subscribe to data	124
10.17	Service information	124
10.18	Service condition	124
11	Test methods and expected results	125
11.1	General	125
11.2	Communication channel security test	125
11.3	Data protection test	126
11.3.1	Data Compression and packaging	126
11.3.2	Data authentication and signature	126
11.3.3	Encryption	126
11.3.4	Digital signature test	126
11.4	SECOM ship/shore test	126
11.4.1	General	126

11.4.2	Prerequisites SECOM ship/shore EUT	129
11.4.3	Upload data.....	129
11.4.4	Download data	130
11.5	SECOM Information Service test	131
11.5.1	General	131
11.5.2	Prerequisites SECOM information service EUT	132
11.5.3	Access	133
11.5.4	Access notification	133
11.5.5	Acknowledgement	134
11.5.6	Capability	134
11.5.7	EncryptionKey	135
11.5.8	EncryptionKey Notification	135
11.5.9	Get.....	136
11.5.10	Get By Link	137
11.5.11	Get Summary	138
11.5.12	Get Public Key	138
11.5.13	Upload Public Key	139
11.5.14	Ping	139
11.5.15	Subscription	139
11.5.16	Subscription Notification	140
11.5.17	Remove Subscription	140
11.5.18	Upload	141
11.5.19	Upload Link	141
11.6	SECOM PKI Service test	142
11.6.1	Prerequisites PKI EUT	142
11.6.2	CRL	142
11.6.3	OCSP.....	143
11.6.4	Revoke.....	143
11.6.5	CSR	144
11.6.6	GetPublicKey	144
11.7	SECOM Service Discovery test.....	144
11.7.1	General	144
11.7.2	Prerequisites Service Discovery EUT	145
11.7.3	Search service - By geometry.....	145
11.7.4	Search service - Without specified search criteria	145
Annex A (normative)	REST service interface definitions	147
A.1	Purpose.....	147
A.2	SECOM information service REST interface definition	147
A.3	SECOM PKI service REST interface definition	147
A.4	SECOM discovery service REST interface definition	147
Annex B (informative)	Operational use cases and profiles	148
B.1	Purpose.....	148
B.2	Use cases and service interface profiles	148
B.2.1	UC-1 Ship shares route plan with service providing enhanced monitoring	148
B.2.2	UC-2 Pilot routes	149
B.2.3	UC-3 Route optimization	150

B.2.4	UC-4 Enhanced monitoring service requests route plan from/for ship for monitoring	151
B.2.5	UC-5 Discover service instance to consume	152
B.2.6	UC-6 Chart (ENC) updates	153
B.2.7	UC-7 navigational warning service	154
B.2.8	UC-8 Updates for detailed bathymetry and tidal and water level forecasts	155
Annex C (informative) Message exchange patterns		157
C.1	Purpose.....	157
C.2	Message exchange pattern.....	157
C.2.1	Generic message exchange patterns.....	157
C.2.2	Alternative and error sequences	160
Annex D (informative) Guidance on implementation.....		161
D.1	Purpose.....	161
D.2	On ship.....	162
D.3	On shore	163
D.4	Service composition	164
D.5	Private side security	165
D.6	SECOM PKI.....	165
D.6.1	General	165
D.6.2	Specification	166
D.6.3	Identity Management.....	166
D.6.4	Public Key Infrastructure	169
D.6.5	Decentral PKI.....	170
D.7	SECOM Service discovery	175
D.7.1	Example 1: geometry combined with serviceType search	175
D.7.2	Example 2: Search with AND condition.....	175
Annex E (informative) Use of white list		176
E.1	Purpose.....	176
E.2	Authorization to access data.....	176
E.3	Access control list	177
E.4	Authorization based on predefined rules or list	177
E.5	Manually updated list.....	178
E.6	Rule based handling on request to information (rule based authorization).....	178
E.7	Rule based request for information	178
E.8	Procedure when receiving "Not authorized"	178
Annex F (informative) Test and simulators.....		179
F.1	Purpose.....	179
F.2	Manual testing	179
F.3	Ship and shore equipment	179
F.4	SECOM information service equipment.....	180
F.5	SECOM PKI equipment	180
F.6	SECOM Service Discovery equipment	181
Bibliography		182
Figure 1 – Overview of SECOM		22
Figure 2 – Secure communication channel		24

Figure 3 – Illustration of what parts of the message are protected by the two signatures	25
Figure 4 – Envelope and data validation	26
Figure 5 – Service definition model for the service interface definitions	28
Figure 6 – Example in C# of conversion from PEM format to minified public key	32
Figure 7 – Example of a public key in PEM format converted to a single line string ...	33
Figure 8 – Example in C# of conversion from minified public key to PEM format	34
Figure 9 – Example of a minified public key string restored to the original PEM format 34	
Figure 10 – UUID version and variant	38
Figure 11 – Upload interface UML diagram	41
Figure 12 – Sequence diagram for upload signed unclassified data with acknowledgement.....	45
Figure 13 – Update link interface UML diagram	46
Figure 14 – Sequence diagram for Upload link to large data	49
Figure 15 – Acknowledgement interface UML diagram.....	51
Figure 16 – Sequence diagram for Acknowledgement interface	54
Figure 17 – Get interface UML diagram	55
Figure 18 – Sequence diagram for Get interface	58
Figure 19 – Sequence diagram for Get interface and classified data	58
Figure 20 – Get Summary interface UML diagram	59
Figure 21 – Sequence diagram for Get Summary interface	62
Figure 22 – Get By Link interface in UML	62
Figure 23 – Sequence diagram for Get By Link interface	64
Figure 24 – Access interface UML diagram.....	65
Figure 25 – Sequence diagram for Request Access and Access Notification interface 67	
Figure 26 – Access Notification interface UML diagram	68
Figure 27 – Subscribe interface UML diagram	70
Figure 28 – Sequence diagram for Subscribe interface	72
Figure 29 – Operational sequence diagram for Subscription interfaces	72
Figure 30 – Sequence diagram for Subscription interfaces with external subscription request.....	73
Figure 31 – Remove Subscription interface UML diagram	74
Figure 32 – Sequence diagram for Remove Subscription interface	75
Figure 33 – Subscription Notification interface UML diagram	76
Figure 34 – Sequence diagram for Subscription Notification interface	77
Figure 35 – Capability interface UML diagram	79
Figure 36 – Sequence diagram for Capability interface	81
Figure 37 – Ping interface UML diagram.....	82
Figure 38 – Check status on service	83
Figure 39 – Encryption Key interface UML diagram	84
Figure 40 – Operational sequence diagram for EncryptionKey upload interface	88
Figure 41 – Operational sequence diagram for EncryptionKey notification interface...	89

Figure 42 – PublicKey interface UML diagram	90
Figure 43 – Operational sequence diagram for PublicKey interface	92
Figure 44 – Principle for service authentication	94
Figure 45 – Sequence for SECOM encryption key management	101
Figure 46 – Alternative sequence for SECOM encryption key management	101
Figure 47 – CSR interface UML diagram.....	106
Figure 48 – Operational sequence diagram for CSR	108
Figure 49 – GetPublicKey interface UML diagram	108
Figure 50 – Operational sequence diagram for GetPublicKey	110
Figure 51 – GetCRL interface UML diagram	110
Figure 52 – Operational sequence diagram for CRL	111
Figure 53 – GetOCSP interface UML diagram.....	111
Figure 54 – Operational sequence diagram for OCSP	113
Figure 55 – PostRevoke interface UML diagram	113
Figure 56 – Operational sequence diagram for Revoke.....	115
Figure 57 – Search service UML information diagram	116
Figure C.1 – Message Exchange Pattern - ONE_WAY.....	157
Figure C.2 – Message Exchange Pattern - REQUEST_CALLBACK	158
Figure C.3 – Message exchange pattern - REQUEST_RESPONSE	158
Figure C.4 – Message exchange pattern - PUBLISH_SUBSCRIBE (Provider nominates)	159
Figure C.5 – Message exchange pattern - PUBLISH_SUBSCRIBE (Consumer request) 159	
Figure C.6 – Error sequence; Incorrect uploaded message	160
Figure C.7 – Error sequence; Unauthorized upload of message	160
Figure C.8 – Error sequence; Unauthorized subscription request.....	160
Figure D.1 – Overview of SECOM	161
Figure D.2 – Overview of certificate usage	162
Figure D.3 – Deployment example for SECOM on ship	163
Figure D.4 – Deployment example for SECOM on shore	164
Figure D.5 – Service composition	165
Figure D.6 – Certificate renewal strategy	173
Figure D.7 – Response from service registry	175
Figure F.1 – Manual testing	179
Figure F.2 – Overview of test equipment for ship and shore equipment	180
Figure F.3 – Overview of test equipment for SECOM information service equipment 180	
Figure F.4 – Overview of test equipment for SECOM PKI equipment	181
Figure F.5 – Overview of test equipment for SECOM service discovery equipment ..	181
Table 1 – Read instructions for tables in service interface definitions	28
Table 2 – SECOM Service interface versioning.....	30
Table 3 – Basic data types	30

Table 4 – SECOM_ExchangeMetadataObject	31
Table 5 – DigitalSignatureValueObject	32
Table 6 – PaginationObject	34
Table 7 – ContainerTypeEnum	35
Table 8 – SECOM_DataProductType	35
Table 9 – SECOM_ResponseCodeEnum	36
Table 10 – AckRequest Enum.....	36
Table 11 – Common HTTP codes	36
Table 12 – Supported WKT geometric objects	37
Table 13 – UUID variants	38
Table 14 – UUID versions.....	38
Table 15 – Service interfaces overview.....	39
Table 16 – Information input for Upload interface	41
Table 17 – Information output for Upload interface	42
Table 18 – REST implementation of Upload.....	43
Table 19 – HTTP Response codes and message in response object	43
Table 20 – Information input for Upload Link interface	47
Table 21 – Information output for Upload Link interface	48
Table 22 – REST implementation of Upload Link	48
Table 23 – HTTP Response codes and message in response object	48
Table 24 – Information input for Acknowledgement interface	51
Table 25 – Enumerations for not acknowledged.....	52
Table 26 – Information output for Acknowledgement interface	52
Table 27 – Enumerations for Acknowledgement interface	53
Table 28 – REST implementation of acknowledgement.....	53
Table 29 – HTTP Response codes and response message.....	53
Table 30 – Information input for Get interface	55
Table 31 – Information output for Get interface	56
Table 32 – REST implementation of Get.....	56
Table 33 – HTTP Response code and message of Get	57
Table 34 – Information input for Get Summary interface	60
Table 35 – Information output for Get Summary interface	60
Table 36 – REST implementation of Get Summary	61
Table 37 – HTTP Response codes and messages of Get Summary	61
Table 38 – Information input for Get By Link interface	63
Table 39 – Information output for Get By Link interface	63
Table 40 – REST implementation of Get By Link	63
Table 41 – HTTP Response code and message of Get By Link.....	63
Table 42 – Information input for Access interface	65
Table 43 – Information output for Access interface	66
Table 44 – Enumerations for Access interface	66
Table 45 – Parameter binding for the operation	66

Table 46 – HTTP Response codes	66
Table 47 – Information input for Access Notification interface	68
Table 48 – Information output for Access Notification interface	68
Table 49 – Parameter binding for the operation	68
Table 50 – HTTP response codes	69
Table 51 – Information input for Subscription interface	70
Table 52 – Information output for Subscription interface	71
Table 53 – REST implementation of Subscription	71
Table 54 – HTTP response codes and messages of Subscription	71
Table 55 – Information input for Remove Subscription interface	74
Table 56 – Information output for Remove Subscription interface	74
Table 57 – REST implementation of Remove Subscription	74
Table 58 – HTTP Response codes and messages of Remove Subscription	75
Table 59 – Information input for Subscription Notification interface	76
Table 60 – Information output for Subscription Notification interface	76
Table 61 – Enumerations for Subscription Notification interface	76
Table 62 – Information exchange for Subscription Notification	77
Table 63 – HTTP response codes for Subscription Notification	77
Table 64 – Capability example	78
Table 65 – Information output for Capability interface	79
Table 66 – REST implementation of Capability	80
Table 67 – HTTP response codes and messages of Capability	81
Table 68 – Information output for Ping interface	82
Table 69 – REST implementation of Ping	82
Table 70 – HTTP response codes of Ping	83
Table 71 – Information input for Encryption Key interface	84
Table 72 – Information input for Encryption Key Notification interface	85
Table 73 – Information output for Encryption Key interface	86
Table 74 – REST implementation of EncryptionKey upload	86
Table 75 – HTTP response codes of EncryptionKey upload	87
Table 76 – REST implementation of EncryptionKey notification	87
Table 77 – HTTP response codes of EncryptionKey notification	87
Table 78 – Information input for PublicKey interface	90
Table 79 – Information output for PublicKey interface GET and information input for PublicKey interface POST	90
Table 80 – REST implementation of PublicKey (GET)	91
Table 81 – HTTP response code and message of PublicKey (GET)	91
Table 82 – REST implementation of PublicKey (POST)	91
Table 83 – HTTP response code and message of PublicKey (POST)	92
Table 84 – Conversion rules	97
Table 85 – Interfaces with envelope signature	97
Table 86 – Command examples	99

Table 87 – Example of commands	102
Table 88 – Creation of public and private key pairs - Example of basic commands ..	104
Table 89 – PKI interface overview	106
Table 90 – Information input for CSR interface	106
Table 91 – Information output for CSR interface	107
Table 92 – REST implementation of CSR	107
Table 93 – HTTP response codes and message in response object.....	107
Table 94 – Information input for GetPublicKey interface	108
Table 95 – Information output for GetPublicKey interface	109
Table 96 – REST implementation of GetPublicKey interface	109
Table 97 – HTTP Response codes and message in response object	109
Table 98 – REST implementation of CRL.....	110
Table 99 – HTTP response codes and message in response object.....	111
Table 100 – REST implementation of OCSP.....	112
Table 101 – HTTP response codes and message in response object.....	112
Table 102 – REST implementation of OCSP	112
Table 103 – HTTP response codes and message in response object.....	113
Table 104 – Information input for Revoke interface	114
Table 105 – Enumerations for Revoke interface	114
Table 106 – Information output for Revoke interface.....	114
Table 107 – REST implementation of Revoke.....	114
Table 108 – HTTP response codes and message in response object.....	115
Table 109 – Information input for search service interface	117
Table 110 – Information input for search parameter object	117
Table 111 – Information output for search service interface	117
Table 112 – REST implementation for Search Service.....	118
Table 113 – HTTP response codes.....	118
Table 114 – Test data reference	126
Table 115 – Upload test method steps.....	129
Table 116 – Download test method steps	130
Table 117 – Test data reference	131
Table 118 – Access test method steps	133
Table 119 – Access Notification test method steps	133
Table 120 – Acknowledgement test method steps	134
Table 121 – Capability test method steps	134
Table 122 – EncryptionKey test method steps	135
Table 123 – EncryptionKey notification test method steps	136
Table 124 – Get test method steps	136
Table 125 – Get By Link test method steps.....	137
Table 126 – Get Summary test method steps	138
Table 127 – Get Public Key test method steps.....	138
Table 128 – Upload Public Key test method steps	139

Table 129 – Ping test method steps	139
Table 130 – Subscription test method steps	140
Table 131 – Subscription Notification test method steps	140
Table 132 – Remove Subscription test method steps	141
Table 133 – Upload test method steps	141
Table 134 – Upload Link test method steps	142
Table 135 – CRL test method steps	143
Table 136 – OCSP test method steps	143
Table 137 – Revoke test method steps	143
Table 138 – CSR test method steps	144
Table 139 – GetPublicKey test method steps	144
Table 140 – Search service by geometry test method steps	145
Table 141 – Search service empty query test method steps	146
Table B.1 – UC-1 Ship shares route plan with service providing enhanced monitoring 149	
Table B.2 – Required service interfaces in UC-3	150
Table B.3 – Required service interfaces in UC-3	151
Table B.4 – Required service interfaces in UC-4	152
Table B.5 – Required service interfaces in UC-6	154
Table B.6 – Required service interfaces in UC-7	155
Table B.7 – Required service interfaces in UC-8	156
Table D.1 – Subject distinguished names fields	172
Table D.2 – Suggested defined fields and their usage by entity kinds	172

INTERNATIONAL ELECTROTECHNICAL COMMISSION

**Maritime navigation and radiocommunication equipment and systems -
Data interfaces - Part 2: Secure communication between ship and shore
(SECOM) –****FOREWORD**

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.

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.

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.

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.

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.

All users should ensure that they have the latest edition of this publication.

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.

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.

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

IEC 63173-2 has been prepared by IEC technical committee 80: Maritime navigation and radiocommunication equipment and systems. It is an International Standard.

This second edition cancels and replaces the first edition published in 2022. This edition constitutes a technical revision.

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

- a) update digitalSignature object to be compatible with IHO S-100 Edition 5.2;
- b) add new data product type;
- c) change in the sequence diagram in Upload, Access, Subscription, EncryptionKey interface;
- d) add subscription identifier attribute in Upload, UploadLink interface;
- e) add callback endpoint attribute in Upload, UploadLink, Access, Subscription, EncryptionKey Notification interface;
- f) add digitalSignatureReference attribute in Acknowledgement, EncryptionKey, EncryptionKey Notification interface;
- g) add attribute publicKey in Capability interface;
- h) update the Search interface including pagination object;
- i) update MCP description for SECOM PKI in Annex D.6;
- j) update service discovery example in Annex D.7.

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

Draft	Report on voting
XX/XX/FDIS	XX/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.

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

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

- reconfirmed,
- withdrawn, or
- revised.

INTRODUCTION

E-navigation has been defined as the means of providing electronic information in a harmonized way, and maritime services have been specified by the International Maritime Organization (IMO). The maritime services are operational services for actors both ashore and onboard. To make the maritime services interoperable between different actors and systems from different manufactures standards, specifications and guidelines in several layers are required, for example technical services and data/product formats. Technical services comprises a set of technical solutions and communications means to provide a maritime service. IMO's e-navigation strategy implementation plan (SIP) requires that all maritime services are IHO S-100 conformant as a baseline. Further, IEC is expected to implement the details as outlined in the SIP.

Secure communication between ship and shore (SECOM) provides standards for secure data exchange with technical services. Further, it contains a technical service interface design that is in accordance with the service guidelines and templates defined by IALA and partly included in IHO S-100.

SECOM specifies service interfaces (APIs) for data exchange, data protection measures to enable secure communication and interfaces for service discoverability. SECOM is applicable for IHO S-100 based products but also other data (payload) formats are supported, i.e. SECOM is generally independent of which data type is exchanged.

The standardisation of a common service interface for data exchange will enable wider technical interoperability where the same service interface can be used for exchanging information regardless of its operational use.

Accordingly, the purpose of SECOM is to:

- facilitate standardized information exchange of, for example, IHO S-100 based products part of maritime services such as route plans, nautical chart updates and navigational warnings;
- facilitate interoperability between maritime IT systems;
- reduce the need to support many different (proprietary) service designs;
- utilize the benefits of service oriented architecture in maritime communication, for example to enable ship systems to interact with port systems on the first call to a specific port.

Further reading:

IEC 61174:2015, *Maritime navigation and radiocommunication equipment and systems - Electronic chart display and information system (ECDIS) - Operational and performance requirements, methods of testing and required test results* [IEC 61174:2015 \[1\]](#)

IEC 63173-1, *Maritime navigation and radiocommunication equipment and systems - Data interfaces - Part 1: S-421 route plan based on S-100* [IEC 63173-1 \[2\]](#)

ISO 3166-1:2020, *Codes for the representation of names of countries and their subdivisions — Part 1: Country code* [ISO 3166-1:2020 \[3\]](#)

ISO 7498-2:1989, *Information processing systems — Open Systems Interconnection — Basic Reference Model — Part 2: Security Architecture* [ISO 7498-2:1989 \[4\]](#)

ISO 19162:2019, *Geographic information — Well-known text representation of coordinate reference systems* [ISO 19162:2019 \[5\]](#)

ISO 21188:2018, *Public key infrastructure for financial services — Practices and policy framework* [ISO 21188:2018 \[6\]](#)

ISO/IEC 27000:2018, *Information technology — Security techniques — Information security management systems — Overview and vocabulary* [ISO/IEC 27000:2018 \[7\]](#)

ISO/IEC 9594-8:2020, *Information technology — Open systems interconnection — Part 8: The Directory: Public-key and attribute certificate frameworks* [ISO/IEC 9594-8:2020](#) [8]

ISO/IEC 10118-1:2016, *Information technology — Security techniques — Hash-functions — Part 1: General* [ISO/IEC 10118-1:2016](#) [9]

ISO/IEC 11770-3:2021, *Information security — Key management — Part 3: Mechanisms using asymmetric techniques* [ISO/IEC 11770-3:2021](#) [10]

ISO/IEC 14888 (all parts), *Information technology - Security techniques - Digital signatures with appendix* [ISO/IEC 14888 \(all parts\)](#) [11]

ISO/IEC 21778:2017, *Information technology — The JSON data interchange syntax* [ISO/IEC 21778:2017](#) [12]

IALA G1128:2018, *The Specification of e-Navigation Technical Services*, ed. 1.1 [13]

IALA G1143, *Unique Identifiers for Maritime Resources* [14]

RFC 2045, *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies* [15]

RFC 3986, *Uniform Resource Identifier (URI): Generic Syntax* [16]

RFC 5234, *Augmented BNF for Syntax Specifications: ABNF* [17]

RFC 8141, *Uniform Resource Names (URNs)* [18]

RFC 8259, *The JavaScript Object Notation (JSON) Data Interchange Format* [19]

Roy Fielding, *Architectural Styles and the Design of Network-based Software Architectures* [20]

Unified Modeling Language (UML):2017, *OMG Unified Modeling Language v. 2.5.1* [viewed 2021-10-18]. Available at <https://www.omg.org/spec/UML/> [21]

OASIS, *Reference Model for Service Oriented Architecture 1.0* [22]

NIST, *Computer security resource center* [viewed 2022-02-14]. Available at <https://csrc.nist.gov/glossary/term/nonce> [23]

NIST DSS-FIPS Publication 180-3, *Secure Hash Standard* [24]

NIST DSS-FIPS Publication 202, *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions* [25]

NIST DSS-FIPS Publication 186-3, *Digital Signature Standard (DSS)* [26]

NIST Special Publication (SP) 800-63B, *Digital Identity Guidelines: Authentication and Lifecycle Management* [27]

Maritime navigation and radiocommunication equipment and systems - Data interfaces - Part 2: Secure communication between ship and shore (SECOM) –

1 Scope

The scope of SECOM includes interfaces (APIs) for data exchange (information services), information security measures to enable secure communication and interfaces for service discoverability. SECOM provides technical interoperability, where the same service interface is used for exchanging the information regardless of its operational use, up to the level of exchanging information securely online. Although designed for IHO S-100 based products, SECOM is technically payload agnostic and applicable also for other types of data.

Communication between SECOM information services for data exchange relies on IP based web services. The "last mile" links between a SECOM information service and the end-user application is not defined in this document, thus the communication technology between the vendor API and a ship/shore system can be non-IP based as well as IP based. The informative [Annex D](#) describes one such implementation of this. This allows different solutions between the service and shore/ship's system/applications.

SECOM does not define physical layer or link layer for transport of data between SECOM information services but requires that the transport supports IP communication. SECOM is applicable for both public (governmental) and private (business) services. SECOM is applicable for ship-shore and shore-ship communication and can be used for ship-ship communication.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IHO S-100:2018, IHO Universal Hydrographic Data Model, ed. 4.0.0

RFC 2315, PKCS #7: Cryptographic Message Syntax

RFC 2459, Internet X.509 Public-key infrastructure and attribute certificate frameworks

RFC 2818, HTTP Over TLS (2000)

RFC 2986, PKCS #10: Certification Request Syntax Specification

RFC 4122, A Universally Unique Identifier (UUID) URN Namespace

RFC 5246, TLS version 1.2 (2008)

RFC 5280, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile

RFC 6960, X.509 Internet Public Key Infrastructure, Online Certificate Status Protocol - OCSP

RFC 7231, Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content

RFC 8446, TLS version 1.3 (2018)

3 Terms, definitions and abbreviated terms

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>

3.1 Terms and definitions

3.1.1

access control

access authorization and access restriction

Note 1 to entry: Access control refers to all the steps that are taken to selectively authorize and restrict entry, contact, or use of [assets](#). Access authorizations and restrictions are often established in accordance with business and security requirements.

[SOURCE: [ISO/IEC 27000:2018](#) [7], 3.1, modified - The definition has been rephrased, and a note to entry added.]

3.1.2

actor

role played by a user or any other system that interacts with the subject

[SOURCE: Unified Modeling Language:2017 [21], 18.2.1.1]

3.1.3

authentication

process to confirm that a claimed characteristic of an entity is actually correct

Note 1 to entry: To authenticate is to verify that a characteristic or attribute that appears to be true is in fact true.

[SOURCE: [ISO/IEC 27000:2018](#) [7], 3.5, modified - The definition has been reformulated, and the note to entry added.]

3.1.4

authentication data

information used to verify the claimed identity of an entity, such as an individual, defined role, corporation or institution

[SOURCE: [ISO 21188:2018](#) [6], 3.4]

3.1.5

authorization

granting of rights, which includes the granting of access based on access rights

[SOURCE: [ISO 7498-2:1989](#) [4], 3.3.10]

3.1.6

certificate

public key and identity of an entity ([authentication data](#)), together with some other information, rendered unforgeable by signing the certificate information with the private key of the certifying authority that issued that public key certificate

[SOURCE: [ISO 21188:2018](#) [6], 3.8]

3.1.7

data client

consumer of data

EXAMPLE A vessel.

3.1.8

data server

provider of data

EXAMPLE Navigational warning centre.

3.1.9**digital signature**

data appended to, or cryptographic transformation of, a data unit that allows the recipient of the data unit to prove the source and integrity of the data unit and protect against forgery e.g. by the recipient

[SOURCE: [ISO 7498-2:1989 \[4\]](#), 3.3.26]

3.1.10**digital signature algorithm**

DSA

3.2 Generic name of digital signature algorithm in this document**3.2.1****encryption key**

cryptographic secret key used with symmetric cryptographic techniques

3.2.2**hash**

string of bits which is the output of a function that calculates a value of the input bytes according to some algorithm, for example checksum calculation, compression, SHA-256

Note 1 to entry: The literature on this subject contains a variety of terms that have the same or similar meaning as hash-code. Modification detection code, manipulation detection code, digest, hash-result, hash-value and imprint are some examples.

Note 2 to entry: NIST SP 800-63B uses "message digest" for "hash".

[SOURCE: [ISO/IEC 10118-1:2016 \[9\]](#), 3.3, modified - The definition has been completed, and Note 2 to entry added.]

3.2.3**identity registry**

registry that holds the identities and bindings to public keys

3.2.4**integrity**

<information security> protection of the accuracy and completeness of information

[SOURCE: [ISO/IEC 27000:2018 \[7\]](#), 3.36, modified - The definition has been rephrased.]

3.2.5**javascript object notation****JSON**

lightweight, text-based, language-independent syntax for defining data interchange formats

[SOURCE: IEC 21778:2017]

3.2.6**private key**

cryptographic key of an entity's asymmetric key pair which can only be used by that entity

[SOURCE: [ISO/IEC 11770-3:2021 \[10\]](#), 3.32, modified - The definition has been reformulated, and the note to entry added.]

3.2.7**public key**

cryptographic key of an entity's asymmetric key pair which can be made public

[SOURCE: [ISO/IEC 11770-3:2021 \[10\]](#), 3.33, modified - The definition has been reformulated and the note to entry deleted.]

3.2.8

public key infrastructure

PKI

structure of hardware, software, people, processes and policies that employs digital signature technology to facilitate a verifiable association between the public component of an asymmetric public key pair with a specific subscriber that possesses the corresponding private key

Note 1 to entry: The public key may be provided for digital signature verification, authentication of the subject in communication dialogues, and/or for message encryption key exchange or negotiation

[SOURCE: [ISO 21188:2018 \[6\]](#) 3.48]

3.2.9

representational state transfer

REST

[software architectural](#) style that defines a set of constraints to be used for creating [Web services](#)

Note 1 to entry: Web services that conform to the REST architectural style, called RESTful Web services (RWS), provide interoperability between computer systems on the [Internet](#)

[SOURCE: [Roy Fielding - Representational State Transfer \(REST\)](#)]

3.2.10

route plan exchange format

RTZ

format based on standardizing a route plan, which consists of waypoints where each waypoint contains information related to the leg from the previous waypoint

Note 1 to entry: The route exchange format is a file - RTZ - containing an XML coded version of the route plan.

[SOURCE: IEC 61174:2015]

3.2.11

scheme administrator

SA

body solely responsible for maintaining and coordinating the protection scheme, by controlling membership of the scheme using the identity registry to ensure that all participants operate according to defined procedures

Note 1 to entry: The SA maintains the top level digital root certificate used to operate the protection scheme and is the only body that can certify the identity of the other participants of the scheme. The SA is responsible for distributing the certificate directly to all registered users participating in the protection scheme.

3.2.12

service

mechanism to enable access to one or more capabilities, where the access is provided using a prescribed interface and is exercised consistent with constraints and policies as specified by the [service description](#)

[SOURCE: [OASIS Reference Model for Service Oriented Architecture 1.0](#)]

3.2.13

service consumer

user which uses service instances provided by service providers

Note 1 to entry: All users within the maritime domain can be service customers, for example ships and their crew, authorities, [VTS](#) stations, organizations (e.g. meteorological), commercial service providers, etc.

[SOURCE: [IALA Service Documentation Guidelines G1128](#)]

3.2.14

service instance

service implementation which can be deployed at different places by same or different service providers

3.2.15**service provider**

user which provides instances of services according to a service specification and service instance description

Note 1 to entry: All users within the maritime domain can be service providers, for example authorities, [VTS](#) stations, organizations (e.g. meteorological), commercial service providers, etc.

[SOURCE: [IALA](#) Service Documentation Guidelines G1128]

3.2.16**service registry**

registry (e.g. database) that holds public descriptions (metadata) of services

3.2.17**nonce**

random or non-repeating value that is included in data exchanged by a protocol, usually for the purpose of guaranteeing the transmittal of live data rather than replayed data, thus detecting and protecting against replay attacks

[SOURCE: NIST, Computer security resource center]

3.3 Abbreviated terms

AES advanced encryption standard

API application program interface

CA certificate authority

CBC cipher block chaining

CRL certificate revocation list

CSR certificate signing request

DSS digital signature standard

ENC electronic navigational chart

IALA International Association of Marine Aids to Navigation and Lighthouse Authorities

IHO International Hydrographic Organization

IMO International Maritime Organization

ISO International Organization for Standardization

MRN maritime resource name

OCSP online certificate status protocol

OEM original equipment manufacturer

PEM privacy enhanced mail

PKCS public key cryptography standards

PKI public key infrastructure

SAScheme administrator

SSK self signed key

TLS transport layer security

UML unified modeling language

URI uniform resource identifier

URL uniform resource locator (web address)

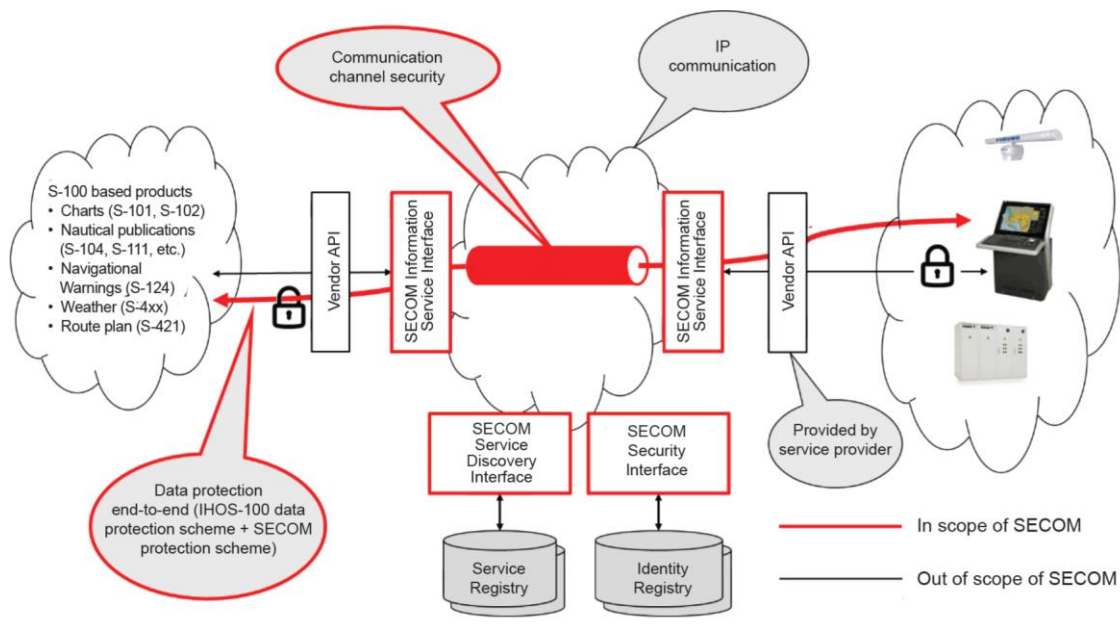
VTS vessel traffic service

WKT well-known text

4 General description of SECOM

4.1 General

SECOM includes information services interfaces (APIs) for data exchange, information security measures by a SECOM [PKI](#), communication channel security and data protection to enable secure communication. Further, SECOM includes interfaces for service discoverability as shown in [Figure 1](#). The purpose with these included components is for SECOM to provide technical interoperability, where the same service interface is used for exchanging the information regardless of its operational use, up to the level of exchanging information securely online. Although designed for [IHO S-100](#) based products, SECOM is technically payload agnostic and applicable also for other types of data.



IEC

Figure 1 – Overview of SECOM

The SECOM information service interface includes the public side exposed on the Internet as depicted in [Figure 1](#). The "last mile" links between a SECOM information service interface and the end-user application is not defined in this document and hence different solutions between the service instance and shore/ship's system are possible. The informative [Annex D](#) describes one such implementation.

SECOM information security contains communication channel security, a variant of [PKI](#) (public key infrastructure) and data protection scheme alternatives for the information exchange with full or partial compliance with [IHO S-100](#). The data protection is provided between end-users. SECOM [PKI](#) includes the definition of a set of service interfaces for key management.

The service discovery interface includes operations to search for service instances from a service registry to meet some criteria, for example chart updates, navigational warnings, updated estimated time of arrival (ETA) information or route optimization services. The service discovery interface allows the user to choose a service instance to consume.

The information exchange between actors is bi-directional, which means that both the ship/shipping company as well as shore authorities and private service providers can initiate the information exchange and act as information provider. SECOM is thus applicable for both ship-to-shore and shore-to-ship communication. Guidance for implementation is available in [Annex D](#).

4.2 Information service interface

The standardisation of service interfaces for information exchange enables a user that implemented the interface, for example for exchange of S-421 Route Plan (IEC 63173-1), to consume a set of operational services with different purpose, but all based on the exchange (push) of S-421 Route Plan. The standard service interface also contains interfaces for subscription, access request, pulling and dynamic request for the instance capability, status and description. See [Annex B](#) for the detailed need for each interface and operation.

Implementing a standardized interface creates significant advantages for a ship as opposed to being forced to implement a vast number of different interfaces in order to adhere to proprietary interfaces from various service providers and consumers. Allowing manufacturers of maritime equipment, such as ECDIS manufacturers, to focus on a standardised service interface will facilitate the implementation of information products onboard ships and generate less requirements for changes onboard when new services are introduced.

Service providers will have similar benefits in producing services based on exchange of standard information products, hereby gaining technical interoperability with all ships having implemented the standard service interface. Since most SECOM interfaces are optional, a service provider of a SECOM information service implementation can decide and pick what to support and hence, enable the possibility to tailor the service to its business operational needs. Each SECOM information service capability can be retrieved from the SECOM information service itself.

The SECOM information service interface is designed for [IHO S-100](#) based products but does also support arbitrary payloads using the SECOM data protection scheme for the exchange. The data can be sent either as open and signed or encrypted and signed.

4.3 Information security

4.3.1 Measures

SECOM contains several measures to meet the following quality attributes.

- Authentication is a process by which a system verifies the identity of a user (human or machine) who wishes to access it (i.e. to confirm, you are who you claim to be, like a passport). This corresponds to the written signature and official stamp on paper. Authentication is achieved in SECOM on two levels: 1) data authentication through the signature (SECOM data protection), and 2) service authentication through the communication channel security and X.509 (RFC 5280 and RFC 2459) certificates (SECOM communication channel security). Authentication is dependent on unique identities which is achieved by SECOM [PKI](#).
- Integrity is achieved in SECOM by signing data where the checksum is part of the signature, described in SECOM data protection. The signing and verification of the signature is dependent on unique identities which is achieved by SECOM [PKI](#).
- Confidentiality is achieved in SECOM on two levels: 1) data encryption according to SECOM data protection and [IHO S-100](#), and 2) secure communication channel (transport security) for IP and HTTP through [TLS](#) and X.509 certificates, described in SECOM communication channel security.
- Access control uses authorization which is the process of determining a set of permissions that is granted to a specific trusted identity. It is achieved in SECOM by the information owner authorizing access to chosen identities from the common identity registry, described in SECOM [PKI](#). Only public interfaces are defined in SECOM (e.g. request access).
- Identification is achieved in SECOM by the common registry for unique identities, bindings to asymmetric key pair and standardized [API](#) to SECOM [PKI](#).
- Non-repudiation, i.e. a service intended to protect against the originator's false denial of having created the content of a message and of having sent a message, is achieved in SECOM by signing the data with the private key.

4.3.2 SECOM PKI

SECOM PKI (public key infrastructure) describes the common interface for key management and the expected functionality to provide the binding between identity and key used for signing data and authentication in service interaction.

A SECOM PKI can be provided by any internationally recognized PKI provider complying with the SECOM PKI requirements. A SECOM PKI provides an interface and an identity registry governed by its scheme administrator. There can be several instances of SECOM PKI provided by multiple scheme administrators. An actor can participate/be registered in several instances of SECOM PKI. SECOM compatible equipment are expected to support multiple instances of SECOM PKI.

SECOM does not specify which instance of identity registry to use, and SECOM does not define or constrain number of identity registries.

4.3.3 Communication channel security

When consuming service instances according to SECOM, the Internet transport is protected with TLS and valid certificates from trusted party as stated in SECOM communication channel security. The protection of the channel is a complement to the protection of the data itself and is necessary to be included to secure also other service requests, such as subscription request, access request and notifications. The SECOM communication channel security describes the usage of certificate obtained from a trusted identity registry and thereby enables authentication on the service interaction itself as depicted in Figure 2.

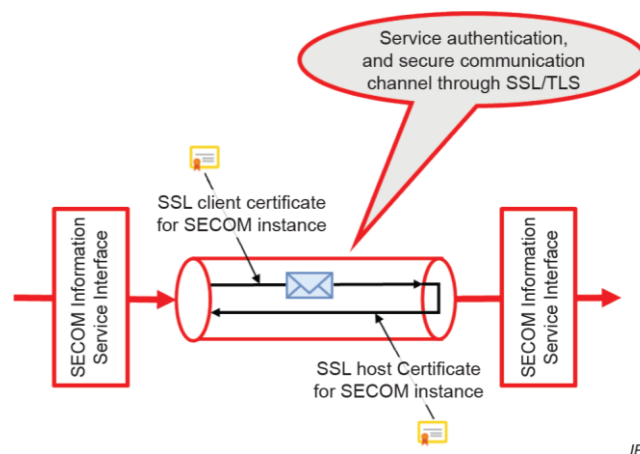


Figure 2 – Secure communication channel

The SECOM communication channel security relies on a SECOM public key infrastructure (SECOM PKI) or public-private key management using X.509 certificates to exchange the keys.

The SECOM communication channel security scheme does not comprise the "last mile" links between the SECOM information service interface and the end-user application. In Annex D, there are informative examples and guidance of how such protection could be achieved.

4.3.4 Data protection

Data protection includes both signing of data for authentication and integrity (digital signature), and optionally encryption of data for confidentiality. Data protection is aligned with the IHO data protection scheme (IHO S-100 part 15).

The digital signature contains both the checksum used to check integrity of received data, and claimed identity for data authentication. The verification of the signature includes access to common public key infrastructure (PKI) where the public key for claimed identity can be downloaded. Whenever data is exchanged, the digital signature shall always be attached and

verified as close to the end user as possible. This gives the basic data end-to-end protection, see [Figure 1](#).

The optional encryption of data uses a common protection scheme and offers two alternatives for encryption key management: 1) the existing encryption key management with permits for [ENC](#) data, such as [IHO](#), can be used with SECOM information service interface and 2) SECOM also includes its own encryption key management that is specially designed for more dynamic data exchange ship to shore and shore to ship, such as route plan exchange. In the SECOM data protection scheme, the sender of classified data creates a temporary encryption key which is transferred securely before the encrypted data is transferred. The temporary encryption key is encrypted with the receiver's public key, the same key as used for authentication, thus can only be decrypted by the actor having the private key in its possession. The receiver's public key if not already present, can be retrieved from the receiver's SECOM interface PublicKey, see [5.7.16](#), or from the SECOM [PKI](#) interface GetPublicKey, see [8.6.3](#). This requires a strong key pair long enough for encrypting and decrypting the encryption key.

The data protection described above only protects the data part in the message. To secure the complete message, an additional signature is added, an envelope signature that includes all information of the envelope contained in the uploadObject, including the data and its dataSignature. The parts of the message that are protected by the different signatures are illustrated in [Figure 3](#).

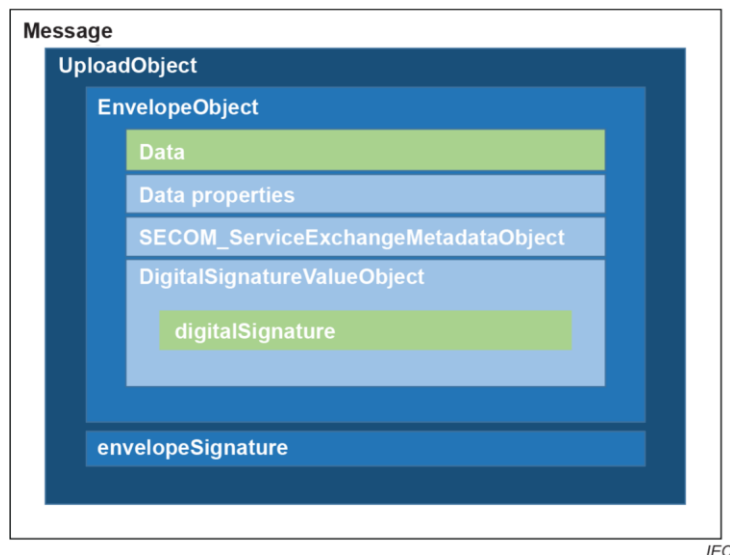


Figure 3 – Illustration of what parts of the message are protected by the two signatures

Another rationale for introducing an envelope signature is to facilitate complete message integrity check at the receiving SECOM information service side thus prohibiting forwarding of corrupt messages during the last mile to the receiver, see [Figure 4](#).

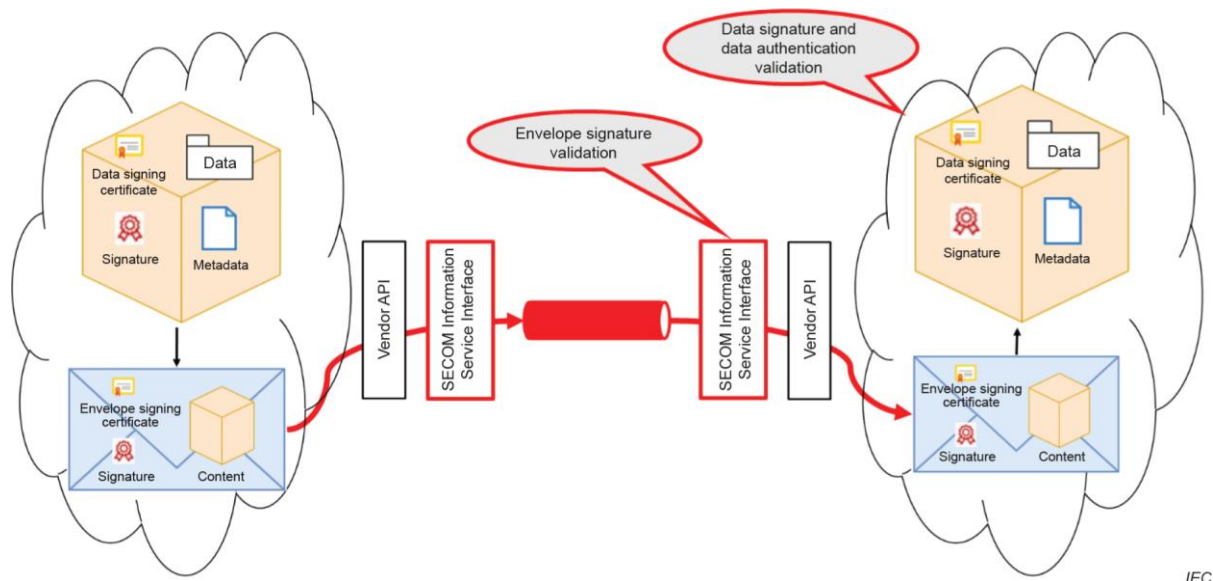


Figure 4 – Envelope and data validation

The SECOM data protection scheme relies on a SECOM public key infrastructure (SECOM PKI) using X.509 certificates to exchange the keys. The private key is generated by the actor and stored internally while the public key is securely uploaded to a SECOM PKI and attached to the identity.

The data protection scheme and the thumbprint of the root certificate indicates the identity registry used in the specific data exchange. The trusted root certificates identified with their thumbprints are expected to be stored in a local trust store.

4.3.5 Certificate revocation status

A certificate revocation list (CRL) is a list of revoked certificates that is downloaded from the certificate authority (CA). Online certificate status protocol (OCSP) is a protocol for checking revocation of a single certificate interactively using an online service called an OCSP responder (see 8.5).

4.4 Service discoverability

To support dynamic use of services, a SECOM service discovery interface has been defined. The intended operational usage is to make it possible for both officers onboard ships and operators ashore to search for service instances to interact with. Some examples from an onboard perspective are a ship that wants to find a provider of route optimization services or a VTS along the intended route to share the route plan with or providers of navigational warnings or electronic navigational charts (ENC). From a shore-based perspective, it is possible to search for registered ships targeted for information requests such as required reporting information, intended route or estimated arrival time. The SECOM service discovery contains common interfaces for discovery of service instances.

A standardized interface for finding service instances reduces the need to support several different proprietary and company specific interfaces. Further, it facilitates increased stability of the implementation, i.e. no new implementation and certification is necessary if a new service registry is used.

SECOM does not specify which service registry to use, and SECOM does not define or constrain the number of service registries, hence there can be several providers of service registries.

SECOM defines only the interface for discovery of service instance, not the registration of service instances in the service registry. This needs to be described by the provider of the chosen service registry.

4.5 Structure of this document

The following clauses are included in SECOM:

- [Clause 5](#) SECOM information service interface;
- [Clause 6](#) SECOM communication channel security;
- [Clause 7](#) SECOM data protection;
- [Clause 8](#) SECOM PKI;
- [Clause 9](#) SECOM service discovery service interface;
- [Clause 10](#) SECOM error cases;
- [Clause 11](#) Test methods and expected results.

5 SECOM information service interface

5.1 General

This [Clause 5](#) describes the SECOM information service interface for exchanging IHO S-100 based products and other payloads. The implementation of the SECOM information service interface supports digital signature and optional data encryption. The SECOM information service interface requires the use of common certificates from the SECOM PKI for authentication between service instances, as described in SECOM communication channel security.

The definition below contains a compilation of several service interfaces that, combined together, constitutes the SECOM information service interface.

In the definition, the expression "consumer" refers to the data client, and the expression "provider" refers to the data server.

- Four of the service interfaces are used for the actual exchange of data; Upload, Upload Link, Get and Get By Link. The Upload service interface is used by providers of information to push data to a consumer with maximum size of 350 kB encoded as Base64 (see RFC 2045). This covers the primary need for exchange of data. The Get interface is used when a consumer pulls data from the provider. For larger amounts of data where the upload (push) cannot be used, the Upload Link interface is used to send a link to the data, and then the data is retrieved with the Get By Link interface.
- The Get Summary interface is used to receive a short summary of available data from where the user can select one or several items to be retrieved by using the Get interface. The Subscription interface is used when a consumer wants to create a subscription to data from the provider. The provider then sends data to the consumer using the Upload service interface. The consumer uses the Remove Subscription interface to delete the subscription. The provider can also "nominate" a consumer by creating a subscription for the consumer using proprietary methods within the provider's system. The provider then uses the Subscription Notification to inform the consumer that there is an active subscription. The provider can also remove a subscription using proprietary methods, and the provider then uses the Subscription Notification to inform the consumer.
- If a consumer does not have access to data, for example retrieval of specific data is not authorized, the consumer uses the Access interface from the provider. The provider then responds with Access Notification to the consumer with the decision, accept or deny.
- Both the provider and consumer use the Capability interface when requesting information regarding accessible interfaces and available types of data. This facilitates for example functions to ask for which version of the S-100 based Product specification a provider or consumer accepts, and functions to only present accessible interfaces for a certain service instance, thus avoiding incorrect use of the service instance.

- The Ping interface is used to check the technical status of the service instance.
- The EncryptionKey interface is used for secure exchange of the symmetric encryption key.
- The PublicKey interface is used for exchanging public certificates used for data encryption and digital signatures. Through this interface, a consumer can both request a certificate (pull) and upload a certificate (push).

5.2 How to read descriptions of service interface definition

The service interface is described according to the service definition model in IHO S-100:2018, part 14 "Online Data Exchange".

The SECOM information service interface constitutes a set of service interfaces, where each service interface is first described as a specification (technology agnostic) and then a technical design of the service interface in REST technology.

Figure 5 describes how to read the service interface as specified in the UML diagram, showing the interface name with the operation(s) defined, input and output. The input and output is defined in the data exchange model. Each interface provides (exposes) at least one service interface (operation) and may require (consume) other service interface(s).

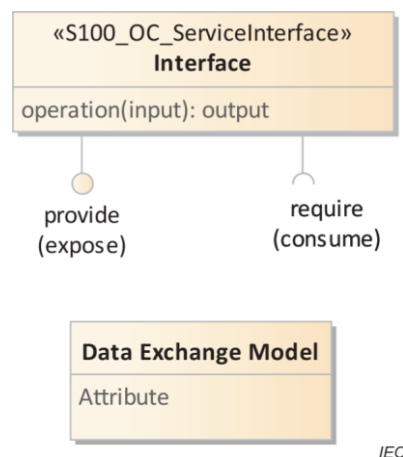


Figure 5 – Service definition model for the service interface definitions

The service specification of the service interface definition describes information for input and output in tables. The multiplicity ("Mult") column describes if data for the attribute is required or not, where 0..1 and 0..* indicates the value may be sent, and 1..* and 1 indicates that value shall be sent (i.e. is required to send). The processing column describes if received data is mandatory or optional to process. If mandatory, the data shall be processed according to the description in dynamic behavior for the service. The processing may include sending asynchronous response. See Table 1.

Table 1 – Read instructions for tables in service interface definitions

<information object>				
Attribute	Mult	Processing	Type	Definition
Name of attribute	This describes for the implementer of the service call (i.e. initial sender of request) if the attribute is required to be sent in the service call or if the attribute may be omitted.	This describes for the implementer of the service call if processing of the attribute by the receiver of sent data (i.e. implementer of the service interface) is mandatory or optional.	This describes the logical (service design independent) encoding of the attribute	The definition and description of the attribute

	Describes for the implementer of the service interface (i.e. responder of initial request) if the attribute is required or not. If an attribute is required by the multiplicity but the attribute is not present, the service call is considered "Bad request" and the service call shall be ignored.	This describes for the implementer of the service interface if processing of the attribute is mandatory or optional. Alternatives of processing: Mandatory Optional		
	Alternatives of multiplicity as defined in UML . 0..1 0..* 1..* 1			

Each service interface also describes a REST service interface design. A table describes each operation in the interface designed according to REST technology. For service interface designed according to REST, the type of data is JSON. See [Annex A](#) for a formal and detailed definition of the whole SECOM information service interface as REST service.

5.3 Service technology and service transportation protocol

The technology (architectural style) chosen is REST (REpresentational State Transfer) upon HTTP/1.1 (RFC 7231).

REST is an architectural style, and an approach to communications that is often used in the development of Web services. The use of REST in SECOM is preferred over other more heavyweight protocols such as SOAP (simple object access protocol) because REST does not leverage as much bandwidth, which makes it a better fit for use in communication between vessels and shore based representation of the same.

REST, which typically runs over HTTP (hypertext transfer protocol), has several architectural constraints.

- Decoupling - Decouples consumers from producers which suits SECOM decentralized architecture well.
- Stateless existence - Also a good prerequisite for a decentralized architecture design.
- Able to leverage a cache - Probably less important in SECOM since most of the interaction is between machines, although for services with man-machine interfaces this is of importance.
- Leverages a layered system - SECOM is dependent on good scaling capabilities which REST supports.
- Leverages a uniform interface - Again since SECOM defines the available services centrally in one or several service registry(s), this constraint supports implementations being decoupled from the services they provide.

The definition of each operation defines additional error messages that the operation specifically shall respond with as a complement to the HTTP response codes defined by HTTP/1.1 (RFC 7231).

5.4 Service interface versioning

The version on the service interface definition follows the version of this document.

The version on the design as REST service interface is reflected in the path for the service instance. The version is defined in the formal REST service definition as OpenAPI specification (Swagger) in [Annex A](#).

[Table 2](#) describes the structure of the versioning of the SECOM service interface.

Table 2 – SECOM Service interface versioning

REST method	URL	base URL with version	SECOM REST operation
POST	https://iec.ch	/v1	/object

EXAMPLE POST <https://iec.ch/v1/object{...}>

NOTE The versioning described here is the version of the SECOM information service interface defined in this document. Version of the service instance itself or other versioning by the manufacturer can be included in the [URL](#).

5.5 Pagination

The Get and Get Summary interfaces may return more than one result. In order to limit the response size, these interfaces support pagination.

When requests to these service interfaces are made, the optional "page" and "pageSize" parameters may be specified by the client. If provided, these allow the client to request how many items are to be returned and the starting point of these items. For example, if a particular request would produce a result consisting of 30 items, setting page = 1 and pageSize = 20 would return the first twenty items; and page = 2, pageSize = 20 would return the remaining ten.

Regardless of whether page or pageSize is specified, the server shall always return the total number of items in the totalItems attribute, and the maximum number of items per page in the maxItemsPerPage attribute.

Errors shall be returned by the server if either an invalid page is asked for or if too many items are requested at a time.

If the page parameter is not present, the server shall send back the first page of the result.

If the pageSize parameter is not present, the server shall send back the maximum number of items per page that it supports.

If the page parameter is set to 0, the server shall not return any items but shall return maxItemsPerPage. This value shall be constant for any query on the server (so that the client only needs to request this once).

5.6 Common information objects and data types

5.6.1 General

The information objects shared among several interfaces are shown in [Table 3](#) to [Table 14](#).

5.6.2 Basic data types

Table 3 – Basic data types

Name	Description
Integer	A signed integer number, the representation of an integer is encapsulation and usage dependent. EXAMPLE 29, -65547
PositiveInteger	An unsigned integer number greater than 0.
Boolean	A value representing binary logic. The value can be either true or false.
CharacterString	A CharacterString is an arbitrary-length sequence of characters including accents and special characters from repertoire of one of the adopted character sets
Date	A date gives values for year, month and day according to the Gregorian Calendar. Character encoding of a date is a string which shall follow the calendar date format (complete representation, basic format) for date. EXAMPLE 1 19980918 (YYYYMMDD)
Time	A time is given by an hour, minute and second in the 24-hour clock system. Character encoding of a time shall be a complete representation of the basic format. Complete representation means that hours, minutes and seconds shall be used. Basic format means that separating characters are omitted.

	Time is preferably expressed as Universal Time Coordinated (UTC). EXAMPLE 2 183059Z Time may be expressed as a Local Time with a given offset to UTC. EXAMPLE 3 183059+0100 Time may be expressed as a Local Time without a specified offset to UTC. EXAMPLE 4 183059 The complete representation of the time of 27 min and 46 s past 15 h locally in Geneva (in winter one hour ahead of UTC), and in New York (in winter five hours behind UTC), together with the indication of the difference between the time scale of local time and UTC, are used as examples. Geneva: 152746+0100 New York: 152746-0500 The service hours for a service that is available all year in an area where Daylight Saving Hour affects the offset to UTC could be expressed as Local Time without specified offset. Opening: 074500 Closing: 161500
DateTime	A DateTime is a combination of a date and a time type. Character encoding of a DateTime shall follow as the above. EXAMPLE 5 19850412T101530
byte[]	Array of bytes

5.6.3 SECOM_ExchangeMetadataObject

The objects shown in [Table 4](#) and [Table 5](#) contain metadata attributes with information of protection, compression and public keys used for the digital signature of the data. The order of adding attributes as part of the generation and verification of the envelope signature is achieved by traversing the attributes in the order first to last, i.e. dataProtection, protectionScheme and so on, see [7.3.4](#).

Table 4 – SECOM_ExchangeMetadataObject

SECOM_ServiceExchangeMetadataObject				
Attribute	Mult	Processing	Type	Definition
dataProtection	1	Mandatory	Boolean	(S-100) Indicates if the data is encrypted. 0 indicates unencrypted data 1 indicates encrypted data
protectionScheme	1	Mandatory	CharacterString	(S-100) Specification or method used for data protection For example S-63, SECOM
digitalSignatureReference	1	Mandatory	CharacterString	(S-100) Specifies the algorithm used to compute digitalSignatureValue For example "ECDSA-384-SHA2"
digitalSignatureValue	1	Mandatory	DigitalSignatureValueObject	Signed Public Key thumbprint plus the digital signature Table 5
compressionFlag	1	Mandatory	Boolean	(S-100) Indicates if data is compressed. 0 indicates uncompressed 1 indicates compressed

Table 5 – DigitalSignatureValueObject

DigitalSignatureValueObject				
Attribute	Mult	Processing	Type	Definition
publicRootCertificateThumbprint	0..1	Optional	CharacterString	Claimed Thumbprint for Signed Root Key (X.509 Certificate)
publicCertificate	1	Mandatory	CharacterString	Public Key (X.509 Certificate), PEM encoded Base64 string contains only leaf certificate
digitalSignature	1	Mandatory	CharacterString	(S100) The digital signature in HEX format as one row, no trailing return/new line

5.6.4 Transfer of public key

5.6.4.1 General

In all cases when the payload includes claimed public keys used, for example, for signatures and classified data, the keys need to be converted in a manner to fit inside a JSON format and to secure interoperability. Certificates in [PEM](#) format contain line feeds and BEGIN/END header/footer. Line feed encoding characters differ among different operational systems (OS) where for example Unix uses only \n for line feed while Windows uses carriage return + line feed characters (\r\n). In order to secure interoperability between different OS machines, a minified public key is introduced to be used inside the different JSON object payloads. More examples on how to embed [PEM](#) encoded keys in payloads can be found in for example [IHO S-100:2018](#), Part 15-7, Data encryption and licensing.

5.6.4.2 Producer

During the build of a payload JSON object and before adding the public key to the same object, the original key is minified into one single line string by:

- Remove all line feed characters
- Remove header -----BEGIN CERTIFICATE-----
- Remove footer -----END CERTIFICATE-----

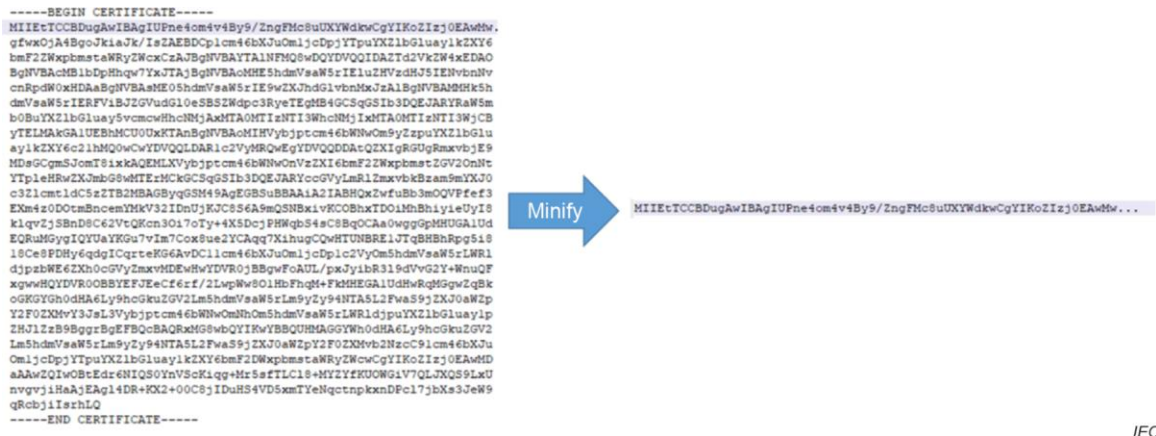
[Figure 6](#) shows an example of the minimisation above using C# .NET. The specific *Environment.NewLine* (the corresponding routine in Java is the *System.lineSeparator()*) is used to ensure that the correct line feed character is used depending on what machine the system is installed on.

```
public string GetMinifiedPemFromCert(X509Certificate2 cert)
{
    var sb = new StringBuilder();
    using (var sw = new StringWriter(sb))
    {
        var writer = new PemWriter(sw);
        writer.WriteObject(DotNetUtilities.FromX509Certificate(cert));
    }
    //1.
    sb.Replace(Environment.NewLine, "");
    //2.
    sb.Replace("-----BEGIN CERTIFICATE-----", "");
    //3.
    sb.Replace("-----END CERTIFICATE-----", "");
    return sb.ToString();
}
```

IEC

Figure 6 – Example in C# of conversion from PEM format to minified public key

Figure 7 shows an example of a public key before and after the conversion.



The diagram illustrates the conversion of a public key from its standard PEM format to a single-line string. On the left, a multi-line PEM block is shown, starting with '-----BEGIN CERTIFICATE-----' and ending with '-----END CERTIFICATE-----'. The body of the certificate is a long string of base64-encoded characters. An arrow labeled 'Minify' points to the right, where the same public key is shown as a single continuous line of base64 characters, with the PEM headers and footers removed.

IEC

Figure 7 – Example of a public key in PEM format converted to a single line string

5.6.4.3 Consumer

For the receiver, the conversion is the opposite. When consuming the transferred payload, the public key is converted back to its original format.

- Add header -----BEGIN CERTIFICATE-----
- Add OS specific line feed character
- Split the public key string from the payload into an array of max 64 characters per row
- For each element in the array
- add element as new row
- add OS specific line feed character
- Add footer -----END CERTIFICATE-----
- Add OS specific line feed character

Figure 8 shows an example of a conversion using C# .NET where the line feeds are added using .NET.

IEC

IEC

maxItemsPerPage	1	Mandatory	PositiveInteger	The maximum number of items the service shall return per page.
-----------------	---	-----------	-----------------	--

5.6.6 ContainerTypeEnum

Container types used in SECOM interface definitions. [Table 7](#) shows the values of the enumeration.

Table 7 – ContainerTypeEnum

ContainerTypeEnum	
Value	Definition
0	S100_DataSet
1	S100_ExchangeSet
2	NONE

5.6.7 SECOM_DataProductType

[Table 8](#) contains the supported product types used in SECOM information interfaces [5.7.5](#), [5.7.6](#), [5.7.8](#), [5.7.10](#) and [5.7.13](#). The manufacturer shall state in user manual which SECOM_DataProductType are supported.

Table 8 – SECOM_DataProductType

SECOM_DataProductType	
Name	Description
OTHER	Other data types not covered in this table
S57	S-57 Electronic Navigational Chart (ENC)
S101	S-101 Electronic Navigational Chart (ENC)
S102	S-102 Bathymetric Surface
S104	S-104 Water Level Information for Surface Navigation
S111	S-111 Surface Currents
S122	S-122 Marine Protected Areas (MPAs)
S123	S-123 Marine Radio Services
S124	S-124 Navigational Warnings
S125	S-125 Marine Navigational Services
S126	S-126 Marine Physical Environment
S127	S-127 Marine Traffic Management
S128	S-128 Catalogue of Nautical Products
S129	S-129 Under Keel Clearance Management (UKCM)
S131	S-131 Marine Harbour Infrastructure
S210	S-210 Inter-VTS Exchange Format
S211	S-211 Port Call Message Format
S212	S-212 VTS Digital Information Service
S401	S-401 Inland ENC
S402	S-402 Bathymetric Contour Overlay for Inland ENC
S411	S-411 Sea Ice Information
S412	S-412 Weather Overlay
S413	S-413 Marine Weather Conditions
S414	S-414 Marine Weather Observations

S421	S-421 Route Plan
route plan exchange format	Route Plan
EPC	Electronic Port Clearance
ASM	Application Specific Message

5.6.8 SECOM_ResponseCodeEnum

The enumeration in [Table 9](#) is added to the synchronous HTTP response object if a validation error occurs at the SECOM instance level for information to the service consumer. This is used in push interfaces such as Upload, UploadLink and Acknowledgement to provide a more specific error message.

Table 9 – SECOM_ResponseCodeEnum

SECOM_ResponseCodeEnum	
Value	Definition
0	Missing required data for the service
1	Failed signature verification
2	Invalid certificate
3	Schema validation error

5.6.9 AckRequest Enum

A flag to indicate that acknowledgement is expected to be returned when the data is delivered to the end user, and/or when the content of the data is processed (opened) by the end user. The value of this enumeration acts as a condition for whether an acknowledgment should be sent (value = 1,2,3) or not (value = 0). It is mandatory to implement the above functionality. [Table 10](#) shows the enumeration values.

Table 10 – AckRequest Enum

AckRequestEnum	
Value	Definition
0 (default)	No ACK requested
1	Delivered ACK requested
2	Opened ACK requested
3	Delivered + Opened ACK requested

5.6.10 Common HTTP response codes

In [Table 11](#), HTTP response codes common to all interfaces optional to implement are listed. For mandatory interfaces such as Ping and Capability, code 501 is not applicable. For interfaces using the SECOM_ResponseCodeEnum in [Table 9](#), the corresponding column value is set to null.

Table 11 – Common HTTP codes

HTTP Code	Message
401	Unauthorized
405	Method not allowed
500	Internal server error
501	Not implemented

5.6.11 Well-known text - WKT

Query parameter "geometry" type used in interface Get, Get Summary and Subscription.

WKT is a human readable representation for spatial objects like points, lines, or enclosed areas on a map. SECOM uses a subset of the objects in **WKT** CRS 2 (see ISO 19162) with EPSG projection 4326 - WGS 84.

The objects used are listed in [Table 12](#).

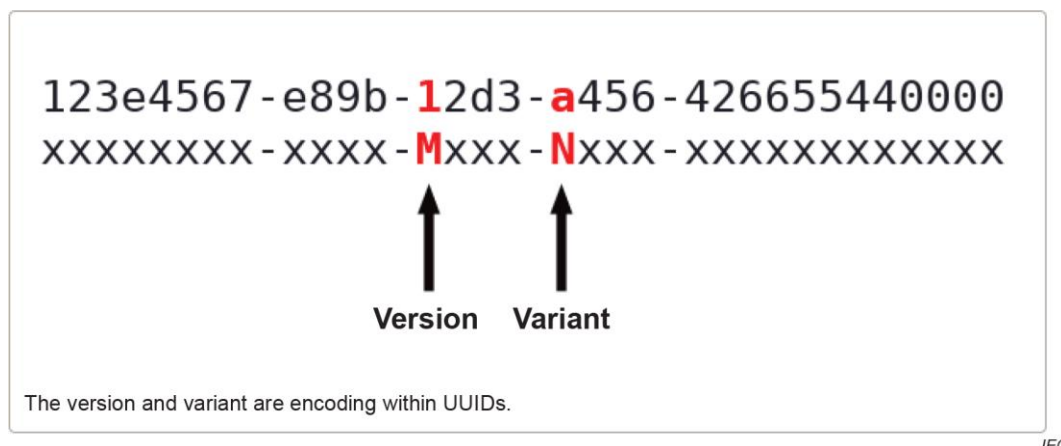
Table 12 – Supported WKT geometric objects

Supported WKT geometric objects		
Object	Definition	Example
Point	A zero-dimensional geometry that represents a single location in coordinate space on a map. The (x,y) coordinate is represented in order (longitude, latitude) and expressed as real value, degree/degree decimal format.	POINT (30 10)
LineString	A curve with linear interpolation between points. Each consecutive pair of points defines a line segment.	LINESTRING (30 10, 10 30, 40 40)
Polygon	A planar surface, i.e. a two-dimensional geometry object, defined by 1 exterior boundary and 0 or more interior boundaries. Each interior boundary defines a hole in the polygon.	POLYGON ((30 10, 40 40, 20 40, 10 20, 30 10))
MultiPoint	An array of point objects	MULTIPOINT ((10 40), (40 30), (20 20), (30 10))
MultiLineString	An array of LineString objects	MULTILINESTRING ((10 10, 20 20, 10 40), (40 40, 30 30, 40 20, 30 10))
MultiPolygon	An array of polygon objects	MULTIPOLYGON (((30 20, 45 40, 10 40, 30 20)), ((15 5, 40 10, 10 20, 5 10, 15 5)))
GeometryCollection	A collection of 1 or more geometry objects	GEOMETRYCOLLECTION (POINT (40 10), LINESTRING (10 10, 20 20, 10 40), POLYGON ((40 40, 20 45, 45 30, 40 40)))

5.6.12 Universally Unique Identifier - UUID

A UUID (see RFC 4122) is a 128-bit (16 bytes) number used to identify some entity in an information system. When generated according to standard methods, UUIDs are unique with a probability of duplication close enough to zero. In SECOM, the UUID type is used as data references and as transaction identifiers created by the information owner in the **OEM** system.

The 128-bit string is represented as 32 hexadecimal digits displayed in five groups separated by hyphens, in the form 8-4-4-4-12 for a total of 36 characters, see [Figure 10](#).



IEC

Figure 10 – UUID version and variant

The four-bit M (13th digit) and the first 1 to 3 most significant bit N (17th digit) fields code the format of the UUID itself. In the example in [Figure 10](#), M = 1 i.e. version 1 and N = a i.e. variant 1 since the only possible hexadecimal values are 8 (1000), 9 (1001), a (1010) or b (1011). There are five versions of UUIDs and four variants where variant 1 (big-endian) is the most common. See [Table 13](#) and [Table 14](#).

Table 13 – UUID variants

UUID Variants			
Name	Binary	Hex Digit	Description
0	0xxx	0-7	Reserved, NCS backwards compatibility
1	10xx	8-b	RFC 4122/DCE 1.1 UUIDs
2	110x	c-d	Reserved, Microsoft Corporation backward compatibility
reserved	111x	e-f	Reserved for future definition

Table 14 – UUID versions

UUID Versions		
Name	Name descr	Description
0	nil	A special case where all bits are set to zero
1	date-time and MAC address	Generated from a time and a node ID
2	date-time and MAC address, security version	Generated from an identifier (usually a group or user id), time, and a node id. RFC 4122 reserves version 2 for "DCE (Distributed Computing Environment) security" UUIDs but without providing any details. For this reason, many UUID implementations omit version-2.
3	name-based	Generated by MD5 (128 bits) hashing of a namespace identifier and name.
4	random	Generated using a random or pseudo-random number.
5	name-based	Generated by SHA-1 (160 bits) hashing of a namespace identifier and name.

The version and variant used when generating the UUID is up to the information owner and hence is outside of SECOM as long as the resulting formatted string complies with the RFC 4122 standard format and can be decoded by the information consumer.

5.6.13 UN/LOCODE

The "United Nations Code for Trade and Transport Locations" is commonly more known as "UN/LOCODE". It is managed and maintained by the UNECE. All details are available from the link below

<https://unece.org/trade/uncefact/unlocode>

Code of location according to UN/Locode codelist restricted to regular expression [a-zA-Z]{2}[a-zA-Z-9]{3}. For example SEGOT, USAA7.

5.7 Service interface definitions

5.7.1 General

Table 15 gives an overview of the service interfaces that constitutes the SECOM information service interface. Service interfaces Capability and Ping shall be supported while the remaining interfaces are optional to implement.

Table 15 – Service interfaces overview

Service Interface	Exchange Pattern	Definition	Use case, see Annex B
Upload	ONE_WAY	Interface to send (push) information to consumer	B.2.1, B.2.2, B.2.3, B.2.4, B.2.7
UploadLink	ONE_WAY	Interface to send (push) a link to information for a consumer to get	B.2.6, B.2.8
Acknowledgement	ONE_WAY	Interface to send acknowledgement on uploaded information.	B.2.1
Get	REQUEST_RESPONSE	Interface to ask for (pull) information from provider	B.2.2
Get Summary	REQUEST_RESPONSE	Interface to ask for (pull) an information list from provider	B.2.2
Get By Link	ONE-WAY	Interface to retrieve information from uploaded link	B.2.6, B.2.8
Access	REQUEST_CALLBACK	Interface to ask for access to information	B.2.1, B.2.4
Access Notification	ONE_WAY	Interface for notification from access request	B.2.1, B.2.4
Subscription	PUBLISH_SUBSCRIBE	Interface to create subscription of information	B.2.1, B.2.4, B.2.7
Remove Subscription	ONE_WAY	Interface to remove subscription	B.2.1, B.2.4, B.2.7
Subscription Notification	ONE_WAY	Interface for notification from subscription events	B.2.1, B.2.4, B.2.7
Capability	REQUEST_RESPONSE	Interface to ask for the interface capabilities. Mandatory to implement.	B.2.1
Ping	REQUEST_RESPONSE	Interface to check status on the service instance. Mandatory to implement.	B.2.4
Encryption Key	ONE_WAY, REQUEST_CALLBACK	Interface to securely send symmetric key for data encryption	B.2.3, B.2.6
PublicKey	ONE_WAY, REQUEST_RESPONSE	Interface to request (pull) and send (push) a public certificate	B.2.3

5.7.2 Service interface - Upload

5.7.2.1 Specification

The purpose of this interface is to upload (push) information that is not larger than a maximum size of 350 kB (Base64 encoded) to an information consumer. An information consumer shall implement this interface in order to receive information while an information provider may implement it.

When uploading (sending) information, acknowledgement can be requested. See interface Acknowledgement for the details of sending acknowledgement.

Uploading (sending) of information can either be within an active subscription of information or uploaded (sent) once as a single message as indicated by the attribute fromSubscription.

The data can be either XML, compressed XML or encrypted binary, typically compressed, as described in the exchange metadata object. The data can contain either one message, such as an S-421 Route Plan (IEC 63173-1) dataset, or a set of messages (files) compressed in a ZIP container and described by an ExchangeSet as described in [IHO S-100](#). The data is encoded as one row Base64 before transferred.

Attached to the data there is a set of metadata describing the protection of the data. The metadata refers to the content in the data attribute, independent of its type. For exchange of an [IHO S-100 DataSet](#), this means that the dataset might be in XML (compressionFlag = false) or in ZIP (compressionFlag = true), or optionally compressed and encrypted (compressionFlag = true and dataProtection = true) and the data is signed (hash calculated and signed). For exchange of an [IHO S-100 ExchangeSet](#) (exchange catalogue), the same procedure is used. The [IHO S-100 ExchangeCatalogue](#) contains a set of files, the folder is compressed, and optionally encrypted and the data is signed (hash calculated and signed).

[Figure 11](#) shows the interface in [UML](#).

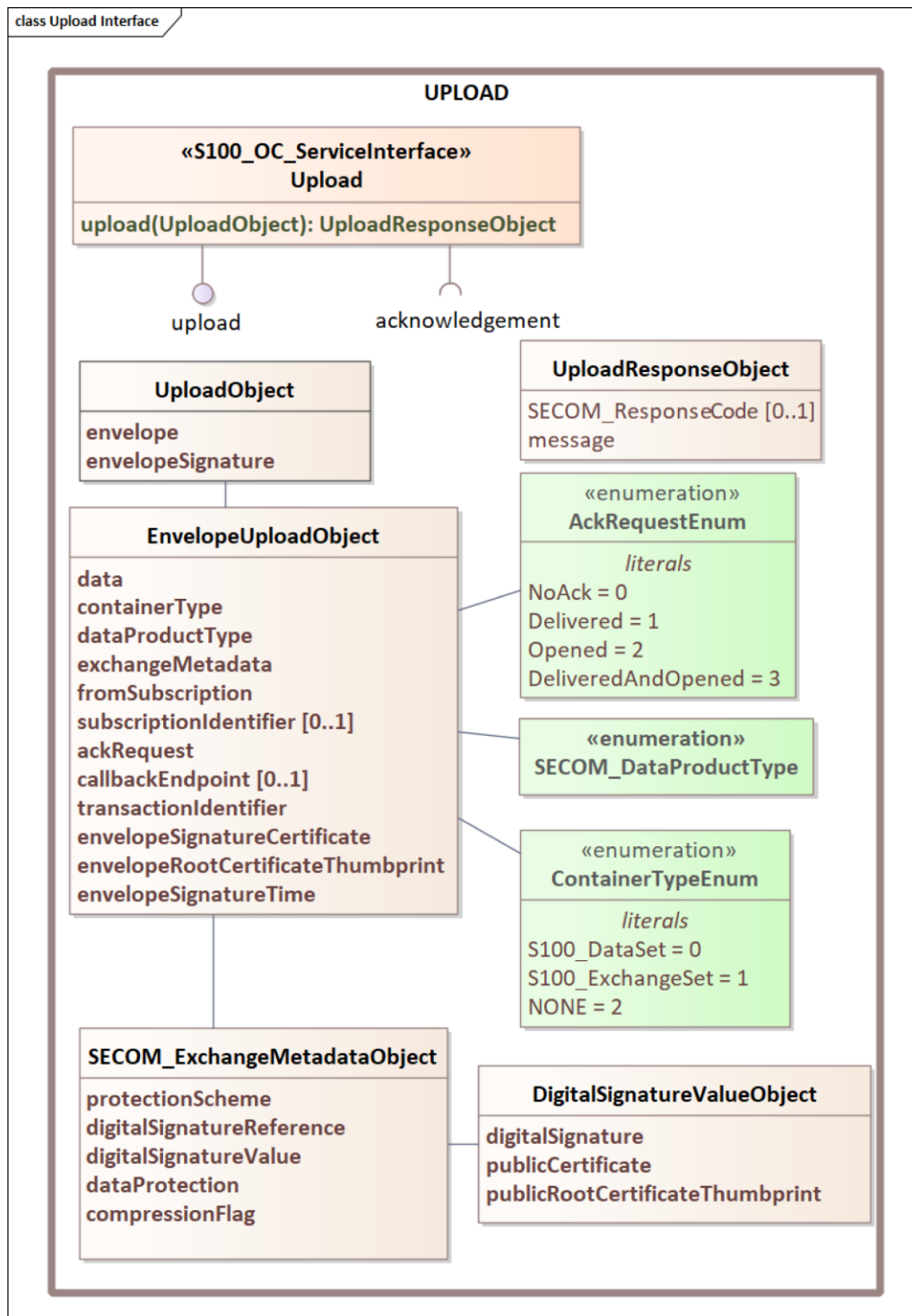


Figure 11 – Upload interface UML diagram

5.7.2.2 Data exchange model

Table 16 and Table 17 describe the data exchanged in the interface.

Table 16 – Information input for Upload interface

UploadObject				
Attribute	Mult	Processing	Type	Definition

envelope	1	Mandatory	EnvelopeUpload-Object	The complete EnvelopeObject being uploaded to receiver including data and message properties
envelopeSignature	1	Mandatory	CharacterString	The signature of the EnvelopeObject in HEX format without whitespace or linebreaks The signature uses same algorithm as defined in digitalSignatureReference
EnvelopeUploadObject				
Attribute	Mult	Processing	Type	Definition
data	1	Mandatory	Base64	The payload XML dataProductType, base64 encoded (e.g. inside a S100_ExchangeSet, S100_DataSet), ZIP or binary. The data can be open, protected and/or compressed.
containerType	1	Mandatory	ContainerType-Enum	Container type of message in the data object Table 7
dataProductType	1	Mandatory	SECOM_Data-ProductType.Name	The name column of the SECOM_DataProductType in Table 8
exchangeMetadata	1	Mandatory	SECOM_Exchange-MetadataObject	The exchange metadata contains information regarding protection scheme, compression, signature and claimed identity according to Table 4
fromSubscription	1	Optional	Boolean	Flag to indicate whether the data has been uploaded within an active subscription or not.
subscriptionIdentifier	0..1	Optional	CharacterString	Subscription identifier is specified if the object is uploaded within subscription.
ackRequest	1	Mandatory	AckRequestEnum	Flag to indicate that acknowledgement is expected to be returned when the data is delivered to the end user, and/or when the content of the data is processed (opened) by the end user.
callbackEndpoint	0..1	Mandatory	CharacterString	URL to the requestor Endpoint where to send an acknowledgement. If not available, the endpoint needs to be available through the search service (see 9.2)
transactionIdentifier	1	Mandatory	UUID	Transaction identifier to be used e.g. in acknowledgement
EnvelopeSignature-Certificate	1	Mandatory	CharacterString	The public certificate of the sender. Used to verify the envelopeObject signature. Contains only leaf certificate.
EnvelopeRoot-CertificateThumb-print	1	Optional	CharacterString	Claimed Thumbprint for Signed Root Key (X.509 Certificate)
EnvelopeSignature-Time	1	Optional	DateTime	Time stamp when the envelope is signed

Table 17 – Information output for Upload interface

UploadResponseObject				
Attribute	Mult	Processing	Type	Definition
SECOM_ResponseCode	0..1	Mandatory	SECOM_ResponseCodeEnum	Additional error code for internal errors Table 9
message	1	Optional	CharacterString	Success or error response message

5.7.2.3 REST design

5.7.2.3.1 General

The service interface and its data exchange model is defined with REST technology.

See [Annex A](#) for formal and detailed definition of the service interface as OpenAPI format.

5.7.2.3.2 Operation POST/object

The interface shall be used for uploading (pushing) data to a consumer. The operation expects one single data object and its metadata. The details are described in [Table 18](#). For detailed description of data exchange model, see [5.7.2.2](#).

Table 18 – REST implementation of Upload

REST Operation			
POST URL/v1/object {body}: return			
REST Parameter (in)	REST Encoding	Mult	Definition
No parameters defined	n/a	n/a	n/a
REST Body (in)	REST Encoding	Mult	Definition
UploadObject	application/json	1	The data (payload) package with its metadata
Return (out)	REST Encoding	Mult	Definition
UploadResponseObject	application/json	1	Confirmation of upload or error message, including description of incorrect or missing values in payload for the specific service.

5.7.2.3.3 Service response

The REST service instance shall respond with HTTP codes and message according to [Table 19](#).

[Table 19](#) describes internal error codes and messages generated by the service implementer. Error codes and messages generated by the deployment environment shall follow RFC 7231 HTTP1/1 and is out of the scope for SECOM.

See also [Table 11](#) for codes common to all interfaces. For this interface with an extra attribute, SECOM_ResponseCodeEnum, this attribute is set to null for the common response codes.

For tests related to these error codes, see [10.14](#).

Table 19 – HTTP Response codes and message in response object

HTTP Code	SECOM_ResponseCodeEnum	Message examples
200	null	Message successfully uploaded
400	0	Missing required data for the service
400	1	Failed signature verification
400	2	Invalid certificate
400	3	Schema validation error
403	null	Not authorized to upload
413	null	Request entity too large

5.7.2.4 Dynamic behaviour

[Figure 12](#) describes the dynamic use of the Upload interface and the relation to the Acknowledgement interface.

The Upload interface is used when data is pushed (sent) to another actor. The size of the data is technically constrained by the chosen REST technology to messages less than 350 kB. In the sequence diagram, the left side actor, SECOM service consumer, produces the data to be pushed to the right side actor. The SECOM service consumer invokes the Upload interface (consumes the Upload service interface) and sends data, data signature and complementary metadata to the receiving actor. The complementary metadata contains attributes for the acknowledgement request (optional), transaction identifier, subscription flag and the signature for the complete transferred data package.

The receiver checks the size of data received (normally by the web server) and will respond with HTTP response 413 if the size is too large for the current configuration. A service consumer shall always be prepared to handle such error response. The receiver of data checks the integrity as part of the verify signature step of the data package and responds accordingly with either HTTP response 200 if OK, or with 400 and SECOM_ResponseCodeEnum = 1 if not OK, see [Table 19](#).

The data is then forwarded to a user application, either directly or through vendor specific communication channels. If delivery acknowledgement has been requested, "delivery ACK" is sent asynchronously to the acknowledgement endpoint appointed in the metadata when the data is correctly forwarded to the end-user.

The end-user application verifies access rights and verifies the signatures as received, both as integrity check and authentication of data received.

If opened acknowledgement has been requested, "opened ACK" shall be sent asynchronously to the acknowledgement endpoint appointed in the metadata when the data is correctly opened or processed by the end-user. SECOM does not specify a time limit between receiving a message and sending an "opened ACK".

Further information on message exchange patterns is given in [Annex C](#).

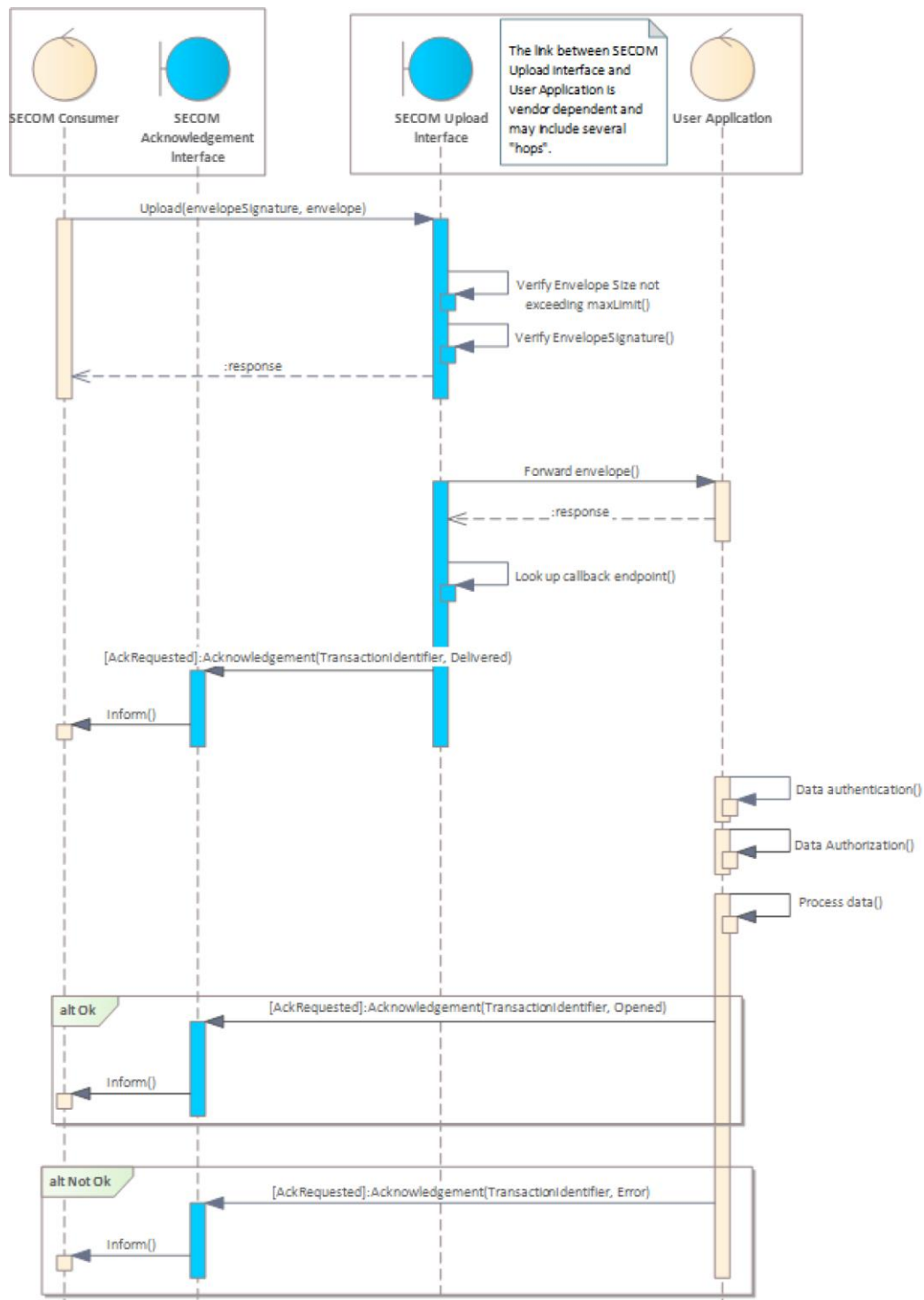


Figure 12 – Sequence diagram for upload signed unclassified data with acknowledgement

NOTE The acknowledgements can be used for supervision and diagnostic purposes.

5.7.3 Service interface - Upload Link

5.7.3.1 Specification

The purpose with this interface is to upload (push) a link to information to a consumer. Hence, a consumer implements this interface in order to receive a link to the information that can be retrieved.

This interface is used when large amounts of data are to be exchanged. The provider of information uploads a link to a consumer, and the consumer then uses the Get by Link interface to pull the data from the provider.

The process of uploading data in a two-step procedure (Upload Link followed by Get By Link) adds a complexity to the signature verification. The transactionIdentifier relates to the actual data but it is still the underlying data that is signed similar to the signing in the Upload interface. Hence, the verification of the data signature can only be done after the data has been downloaded using Get By Link. [Figure 13](#) describes the interface in [UML](#).

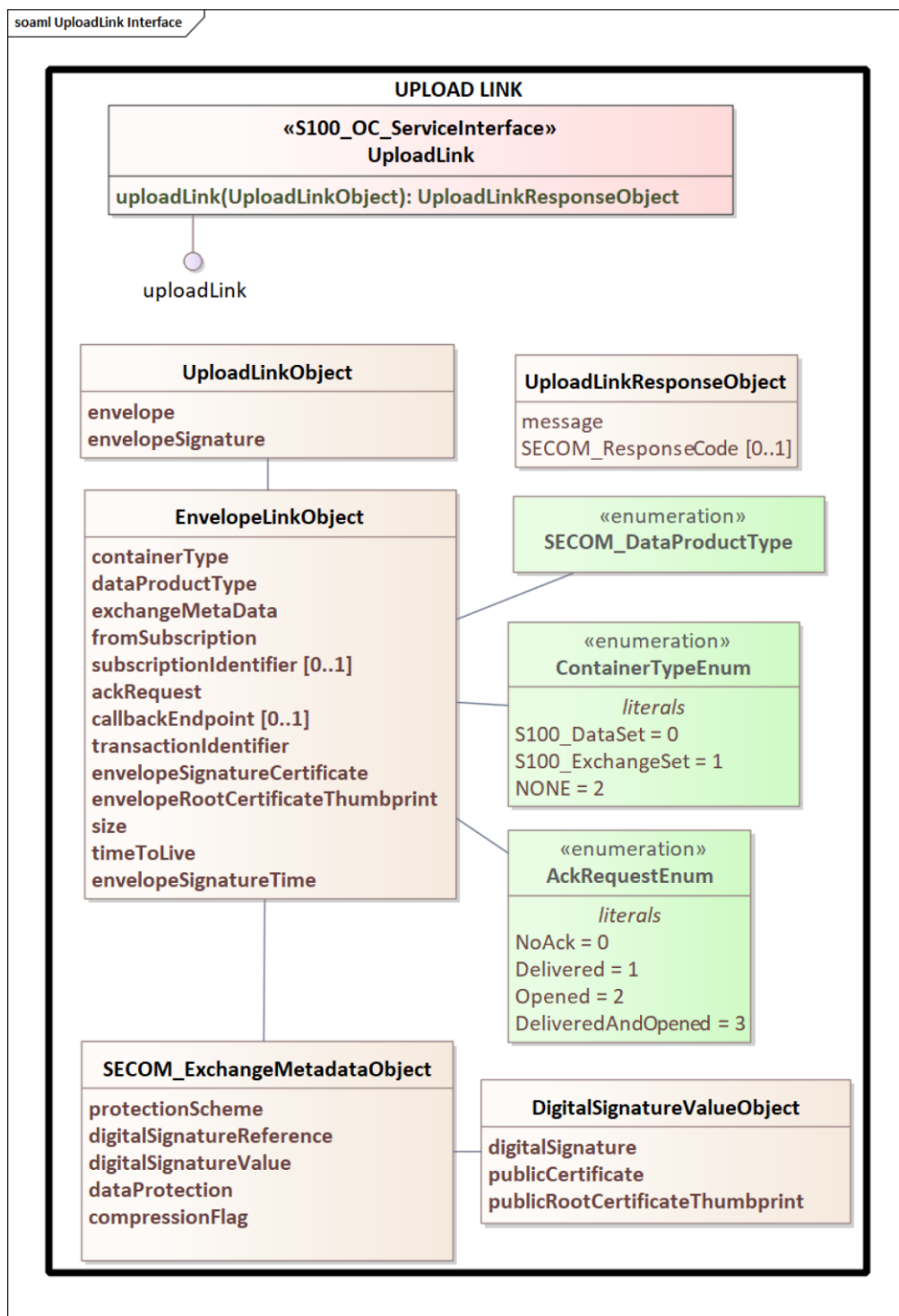


Figure 13 – Update link interface UML diagram

5.7.3.2 Data exchange model

The exchanged data in the service interface is described in detail in [Table 20](#) and [Table 21](#).

Table 20 – Information input for Upload Link interface

UploadLinkObject				
Attribute	Mult	Processing	Type	Definition
envelope	1	Mandatory	EnvelopeLinkObject	The complete EnvelopeLinkObject being uploaded to receiver including message properties
envelopeSignature	1	Mandatory	CharacterString	The signature of the EnvelopeLinkObject in HEX format without whitespace or linebreaks
EnvelopeLinkObject				
Attribute	Mult	Processing	Type	Definition
containerType	1	Mandatory	ContainerTypeEnum	Container type of message in the data object Table 7
dataProductType	1	Mandatory	SECOM_DataProduct-Type.Name	The name column of SECOM_DataProductType in Table 8
exchangeMetadata	1	Mandatory	SECOM_Exchange-MetadataObject	Service exchange metadata with signature, ID etc.
fromSubscription	1	Optional	Boolean	Flag to indicate whether the data has been uploaded within an active subscription or not.
subscriptionIdentifier	0..1	Optional	CharacterString	Subscription identifier is specified if the object is uploaded within subscription.
ackRequest	1	Mandatory	AckRequestEnum	Flag to indicate that acknowledgement is expected to be returned when the data is delivered to the end user, and/or when the content of the data is processed (opened) by the end user.
callbackEndpoint	0..1	Mandatory	CharacterString	URL to the requestor Endpoint where to send an acknowledgement. If not available, the endpoint needs to be available through search service (see 9.2)
transactionIdentifier	1	Mandatory	UUID	Transaction identifier to be used in acknowledgement and when retrieving the message using Get By Link
EnvelopeSignature-Certificate	1	Mandatory	CharacterString	The public certificate of the sender, used to verify the envelopeLinkObject signature. Contains only leaf certificate.
EnvelopeRootCertificate-Thumbprint	1	Optional	CharacterString	Claimed Thumbprint for Signed Root Key (X.509 Certificate)
size	1	Optional	PositiveInteger	Size of the data in kBytes to be downloaded.
timeToLive	1	Mandatory	DateTime	DateTime when data will be deleted on server. The data need to be fetched before this time.
envelopeSignature-Time	1	Optional	DateTime	Time stamp when the envelope is signed

Table 21 – Information output for Upload Link interface

UploadLinkResponseObject				
Attribute	Mult	Processing	Type	Definition
SECOM_ResponseCode	0..1	Mandatory	SECOM_ResponseCodeEnum	Additional error code for internal errors, see Table 9
message	1	Optional	CharacterString	Success or error response message

5.7.3.3 REST Design

5.7.3.3.1 General

The service interface and its data exchange model is defined with REST technology.

5.7.3.3.2 Operation - POST /object/link

The REST operation POST /object/link. The interface shall be used for uploading (pushing) a link to data to a consumer. The details are described in [Table 22](#).

Table 22 – REST implementation of Upload Link

REST Operation			
POST URL /v1/object/link {body}: return			
REST Parameter (in)	REST Encoding	Mult	Definition
No parameters defined	n/a	n/a	n/a
REST Body (in)	REST Encoding	Mult	Definition
UploadLinkObject	application/json	1	The message link with its metadata
Return (out)	REST Encoding	Mult	Definition
UploadLinkResponseObject	application/json	1	Confirmation of upload or error message

5.7.3.3.3 Service response

The service instance shall respond with HTTP codes and message according to [Table 23](#).

See also [Table 11](#) for codes common to all interfaces. For this interface with an extra attribute, SECOM_ResponseCodeEnum, this attribute is set to null for the common response codes.

For tests related to these error codes, see [10.14](#).

Table 23 – HTTP Response codes and message in response object

HTTP Code	SECOM_ResponseCodeEnum	Message
200	null	Link successfully uploaded
400	0	Missing required data for the service
400	1	Failed signature verification
400	2	Invalid certificate
403	null	Not authorized to upload link

5.7.3.4 Dynamic behavior

[Figure 14](#) describes the dynamic behavior of the service interface.

The Upload Link service interface is used when data is larger than technically possible to POST in a REST service call such as Upload. The left side actor in the sequence diagram shall send a larger data package to the user (right side in [Figure 14](#)).