

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**OPC unified architecture –
Part 7: Profiles**

**Architecture unifiée OPC –
Partie 7: Profils**

Withdrawn



THIS PUBLICATION IS COPYRIGHT PROTECTED

Copyright © 2015 IEC, Geneva, Switzerland

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

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

IEC Central Office
3, rue de Varembe
CH-1211 Geneva 20
Switzerland

Tel.: +41 22 919 02 11
Fax: +41 22 919 03 00
info@iec.ch
www.iec.ch

About the IEC

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

About IEC publications

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

IEC Catalogue - webstore.iec.ch/catalogue

The stand-alone application for consulting the entire bibliographical information on IEC International Standards, Technical Specifications, Technical Reports and other documents. Available for PC, Mac OS, Android Tablets and iPad.

IEC publications search - www.iec.ch/searchpub

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 It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished

Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and also once a month by email.

Electropedia - www.electropedia.org

The world's leading online dictionary of electronic and electrical terms containing more than 30 000 terms and definitions in English and French, with equivalent terms in 15 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary

More than 60 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

IEC Customer Service Centre - webstore.iec.ch/csc

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

A propos de l'IEC

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

A propos des publications IEC

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

Catalogue IEC - webstore.iec.ch/catalogue

Application autonome pour consulter tous les renseignements bibliographiques sur les Normes internationales, Spécifications techniques, Rapports techniques et autres documents de l'IEC. Disponible pour PC, Mac OS, tablettes Android et iPad.

Recherche de publications IEC - www.iec.ch/searchpub

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

IEC Just Published - webstore.iec.ch/justpublished

Restez informé sur les nouvelles publications IEC. Just Published détaille les nouvelles publications parues. Disponible en ligne et aussi une fois par mois par email.

Electropedia - www.electropedia.org

Le premier dictionnaire en ligne de termes électroniques et électriques. Il contient plus de 30 000 termes et définitions en anglais et en français, ainsi que les termes équivalents dans 15 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

Glossaire IEC - std.iec.ch/glossary

Plus de 60 000 entrées terminologiques électrotechniques, en anglais et en français, extraites des articles Termes et Définitions des publications IEC parues depuis 2002. Plus certaines entrées antérieures extraites des publications des CE 37, 77, 86 et CISPR de l'IEC.

Service Clients - webstore.iec.ch/csc

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

INTERNATIONAL STANDARD

NORME INTERNATIONALE



**OPC unified architecture –
Part 7: Profiles**

**Architecture unifiée OPC –
Partie 7: Profils**

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

COMMISSION
ELECTROTECHNIQUE
INTERNATIONALE

ICS 25.040.40; 35.100

ISBN 978-2-8322-2372-7

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

CONTENTS

FOREWORD.....	10
1 Scope.....	12
2 Normative references	12
3 Terms, definitions, and conventions.....	13
3.1 Terms and definitions.....	13
3.2 Abbreviations.....	14
4 Overview	14
4.1 General.....	14
4.2 ConformanceUnit	15
4.3 Profiles	15
4.4 Profile Categories	16
5 ConformanceUnits	16
5.1 Overview.....	16
5.2 Services.....	17
5.3 Transport and communication related features.....	28
5.4 Information Model and AddressSpace related features.....	36
5.5 Miscellaneous	55
6 Profiles.....	56
6.1 Overview.....	56
6.2 Profile list	56
6.3 Conventions for Profile definitions.....	62
6.4 Applications	62
6.5 Profile tables.....	64
6.5.1 Introduction	64
6.5.2 Core Server Facet	64
6.5.3 Base Server Behaviour Facet	65
6.5.4 Attribute WriteMask Server Facet	65
6.5.5 File Access Server Facet.....	66
6.5.6 Documentation Server Facet	66
6.5.7 Embedded DataChange Subscription Server Facet.....	66
6.5.8 Standard DataChange Subscription Server Facet	67
6.5.9 Enhanced DataChange Subscription Server Facet.....	67
6.5.10 Data Access Server Facet	68
6.5.11 ComplexType Server Facet.....	68
6.5.12 Standard Event Subscription Server Facet.....	68
6.5.13 Address Space Notifier Server Facet	69
6.5.14 A & C Base Condition Server Facet	69
6.5.15 A & C Address Space Instance Server Facet	70
6.5.16 A & C Enable Server Facet.....	70
6.5.17 A & C Alarm Server Facet.....	70
6.5.18 A & C Acknowledgeable Alarm Server Facet.....	70
6.5.19 A & C Exclusive Alarming Server Facet	71
6.5.20 A & C Non-Exclusive Alarming Server Facet.....	71
6.5.21 A & C Previous Instances Server Facet	71
6.5.22 A & C Dialog Server Facet.....	72
6.5.23 A & E Wrapper Facet.....	72

6.5.24	Method Server Facet	73
6.5.25	Auditing Server Facet	73
6.5.26	Node Management Server Facet.....	73
6.5.27	Client Redundancy Server Facet	74
6.5.28	Redundancy Transparent Server Facet.....	74
6.5.29	Redundancy Visible Server Facet	74
6.5.30	Historical Raw Data Server Facet	75
6.5.31	Historical Aggregate Server Facet	75
6.5.32	Historical Access Structured Data Server Facet.....	76
6.5.33	Historical Data AtTime Server Facet	77
6.5.34	Historical Access Modified Data Server Facet.....	77
6.5.35	Historical Annotation Server Facet.....	77
6.5.36	Historical Data Update Server Facet.....	77
6.5.37	Historical Data Replace Server Facet	78
6.5.38	Historical Data Insert Server Facet	78
6.5.39	Historical Data Delete Server Facet	78
6.5.40	Base Historical Event Server Facet.....	78
6.5.41	Historical Event Update Server Facet.....	79
6.5.42	Historical Event Replace Server Facet.....	79
6.5.43	Historical Event Insert Server Facet.....	79
6.5.44	Historical Event Delete Server Facet	79
6.5.45	Aggregate Subscription Server Facet.....	79
6.5.46	Nano Embedded Device Server Profile	80
6.5.47	Micro Embedded Device Server Profile.....	81
6.5.48	Embedded UA Server Profile	81
6.5.49	Standard UA Server Profile.....	81
6.5.50	Core Client Facet.....	82
6.5.51	Base Client Behaviour Facet.....	82
6.5.52	Discovery Client Facet.....	83
6.5.53	Address Space Lookup Client Facet	83
6.5.54	Entry-Level Support Client Facet	83
6.5.55	Multi-Server Client Connection Facet.....	84
6.5.56	File Access Client Facet	84
6.5.57	Documentation – Client	84
6.5.58	Attribute Read Client Facet.....	84
6.5.59	Attribute Write Client Facet.....	85
6.5.60	DataChange Subscriber Client Facet	85
6.5.61	DataAccess Client Facet.....	85
6.5.62	Event Subscriber Client Facet.....	85
6.5.63	Notifier and Source Hierarchy Client Facet	86
6.5.64	A & C Base ConditionClient Facet	86
6.5.65	A & C Address Space Instance Client Facet	86
6.5.66	A & C Enable Client Facet	87
6.5.67	A & C Alarm Client Facet.....	87
6.5.68	A & C Exclusive Alarming Client Facet.....	87
6.5.69	A & C Non-Exclusive Alarming Client Facet	87
6.5.70	A & C Previous Instances Client Facet.....	88
6.5.71	A & C Dialog Client Facet.....	88
6.5.72	A & E Proxy Facet	

6.5.73	Method Client Facet.....	89
6.5.74	Auditing Client Facet	90
6.5.75	Node Management Client Facet.....	90
6.5.76	Advanced Type Programming Client Facet	90
6.5.77	Diagnostic Client Facet.....	90
6.5.78	Redundant Client Facet	91
6.5.79	Redundancy Switch Client Facet	91
6.5.80	Historical Access Client Facet	91
6.5.81	Historical Annotation Client Facet.....	91
6.5.82	Historical Data AtTime Client Facet	91
6.5.83	Historical Aggregate Client Facet.....	92
6.5.84	Historical Data Update Client Facet	93
6.5.85	Historical Data Replace Client Facet.....	93
6.5.86	Historical Data Insert Client Facet	93
6.5.87	Historical Data Delete Client Facet	93
6.5.88	Historical Access Client Server Timestamp Facet	93
6.5.89	Historical Access Modified Data Client Facet.....	94
6.5.90	Structured Data AtTime Client Facet.....	94
6.5.91	Historical Structured Data Access Client Facet	94
6.5.92	Historical Structured Data Modified Client Facet.....	94
6.5.93	Historical Structured Data Delete Client Facet	94
6.5.94	Historical Structured Data Update Client Facet	95
6.5.95	Historical Structured Data Replace Client Facet	95
6.5.96	Historical Structured Data Insert Client Facet	95
6.5.97	Historical Events Client Facet.....	95
6.5.98	Historical Event Update Client Facet.....	95
6.5.99	Historical Event Replace Client Facet	96
6.5.100	Historical Event Delete Client Facet.....	96
6.5.101	Historical Event Insert Client Facet.....	96
6.5.102	Aggregate Subscriber Client Facet	96
6.5.103	User Token – Anonymous Facet	98
6.5.104	User Token – User Name Password Server Facet	98
6.5.105	User Token – X509 Certificate Server Facet	98
6.5.106	User Token – Issued Token Server Facet	98
6.5.107	User Token – Issued Token Windows Server Facet	98
6.5.108	User Token – User Name Password Client Facet.....	99
6.5.109	User Token – X509 Certificate Client Facet	99
6.5.110	User Token – Issued Token Client Facet	99
6.5.111	User Token – Issued Token Windows Client Facet	99
6.5.112	UA-TCP UA-SC UA Binary.....	99
6.5.113	SOAP-HTTP WS-SC UA XML	100
6.5.114	SOAP-HTTP WS-SC UA Binary	100
6.5.115	SOAP-HTTP WS-SC UA XML-UA Binary	100
6.5.116	HTTPS UA Binary	100
6.5.117	HTTPS UA XML.....	101
6.5.118	Security User Access Control Full.....	101
6.5.119	Security User Access Control Base.....	101
6.5.120	Security Time Synchronization.....	101
6.5.121	Best Practice – Audit Events.....	

6.5.122	Best Practice – Alarm Handling	102
6.5.123	Best Practice – Random Numbers	102
6.5.124	Best Practice – Timeouts	102
6.5.125	Best Practice – Administrative Access	102
6.5.126	Best Practice – Strict Message Handling	103
6.5.127	Best Practice – Audit Events Client.....	103
6.5.128	SecurityPolicy – None.....	103
6.5.129	SecurityPolicy – Basic128Rsa15.....	103
6.5.130	SecurityPolicy – Basic256.....	104
6.5.131	SecurityPolicy – Basic256Sha256.....	104
6.5.132	TransportSecurity – TLS 1.0	105
6.5.133	TransportSecurity – TLS 1.1	105
6.5.134	TransportSecurity – TLS 1.2	105
Bibliography.....		107
Figure 1 – Profile – ConformanceUnit – TestCases		15
Figure 2 – HMI Client sample.....		63
Figure 3 – Embedded Server sample		63
Figure 4 – Standard UA Server sample		64
Table 1 – ProfileCategories.....		16
Table 2 – ConformanceGroups		17
Table 3 – Discovery Services		18
Table 4 – Session Services.....		19
Table 5 – Node Management Services.....		20
Table 6 – View Services.....		21
Table 7 – Attribute Services.....		22
Table 8 – Method Services		23
Table 9 – Monitored Item Services.....		24
Table 10 – Subscription Services.....		27
Table 11 – Security.....		29
Table 12 – Protocol and Encoding		36
Table 13 – Base information		37
Table 14 – Address Space model.....		40
Table 15 – Data Access		42
Table 16 – Alarms and Conditions		43
Table 17 – Historical Access.....		45
Table 18 – Aggregates.....		49
Table 19 – Auditing.....		55
Table 20 – Redundancy		55
Table 21 – Miscellaneous		56
Table 22 – Profile list.....		58
Table 23 – Core Server Facet.....		65
Table 24 – Base Server Behaviour Facet.....		65

Table 25 – Attribute WriteMask Server Facet	66
Table 26 – File Access Server Facet	66
Table 27 – Documentation Server Facet	66
Table 28 – Embedded DataChange Subscription Server Facet	67
Table 29 – Standard DataChange Subscription Server Facet	67
Table 30 – Enhanced DataChange Subscription Server Facet	67
Table 31 – Data Access Server Facet	68
Table 32 – ComplexType Server Facet	68
Table 33 – Standard Event Subscription Server Facet	69
Table 34 – Address Space Notifier Server Facet	69
Table 35 – A & C Base Condition Server Facet	69
Table 36 – A & C Address Space Instance Server Facet	70
Table 37 – A & C Enable Server Facet	70
Table 38 – A & C Alarm Server Facet	70
Table 39 – A & C Acknowledgeable Alarm Server Facet	71
Table 40 – A & C Exclusive Alarming Server Facet	71
Table 41 – A & C Non-Exclusive Alarming Server Facet	71
Table 42 – A & C Previous Instances Server Facet	72
Table 43 – A & C Dialog Server Facet	72
Table 44 – A & E Wrapper Facet	73
Table 45 – Method Server Facet	73
Table 46 – Auditing Server Facet	73
Table 47 – Node Management Server Facet	74
Table 48 – Client Redundancy Server Facet	74
Table 49 – Redundancy Transparent Server Facet	74
Table 50 – Redundancy Visible Server Facet	75
Table 51 – Historical Raw Data Server Facet	75
Table 52 – Historical Aggregate Server Facet	76
Table 53 – Historical Access Structured Data Server Facet	77
Table 54 – Historical Data AtTime Server Facet	77
Table 55 – Historical Access Modified Data Server Facet	77
Table 56 – Historical Annotation Server Facet	77
Table 57 – Historical Data Update Server Facet	78
Table 58 – Historical Data Replace Server Facet	78
Table 59 – Historical Data Insert Server Facet	78
Table 60 – Historical Data Delete Server Facet	78
Table 61 – Base Historical Event Server Facet	79
Table 62 – Historical Event Update Server Facet	79
Table 63 – Historical Event Replace Server Facet	79
Table 64 – Historical Event Insert Server Facet	79
Table 65 – Historical Event Delete Server Facet	79
Table 66 – Aggregate Subscription Server Facet	80
Table 67 – Nano Embedded Device Server Profile	81

Table 68 – Micro Embedded Device Server Profile.....	81
Table 69 – Embedded UA Server Profile	81
Table 70 – Standard UA Server Profile	82
Table 71 – Core Client Facet	82
Table 72 – Base Client Behaviour Facet	83
Table 73 – Discovery Client Facet.....	83
Table 74 – AddressSpace Lookup Client Facet	83
Table 75 – Entry-Level SupportClient Facet	84
Table 76 – Multi-Server Client Connection Facet	84
Table 77 –File Access Client Facet	84
Table 78 – Documentation – Client	84
Table 79 – Attribute Read Client Facet	84
Table 80 – Attribute Write Client Facet	85
Table 81 – DataChange Subscriber Client Facet.....	85
Table 82 – DataAccess Client Facet	85
Table 83 – Event Subscriber Client Facet	86
Table 84 – Notifier and Source Hierarchy Client Facet.....	86
Table 85 – A & C Base Condition Client Facet.....	86
Table 86 – A & C Address Space Instance Client Facet.....	86
Table 87 – A & C Enable Client Facet	87
Table 88 – A & C Alarm Client Facet.....	87
Table 89 – A & C Exclusive Alarming Client Facet	87
Table 90 – A & C Non-Exclusive Alarming Client Facet.....	88
Table 91 – A & C Previous Instances Client Facet	88
Table 92 – A & C Dialog Client Facet.....	88
Table 93 – A & E Proxy Facet.....	89
Table 94 – Method Client Facet	89
Table 95 – Auditing Client Facet.....	90
Table 96 – Node Management Client Facet.....	90
Table 97 – Advanced Type Programming Client Facet	90
Table 98 – Diagnostic Client Facet	90
Table 99 – Redundant Client Facet.....	91
Table 100 – Redundancy Switch Client Facet	91
Table 101 – Historical Access Client Facet	91
Table 102 – Historical Annotation Client Facet.....	91
Table 103 – Historical Data AtTime Client Facet	92
Table 104 – Historical Aggregate Client Facet	92
Table 105 – Historical Data Update Client Facet	93
Table 106 – Historical Data Replace Client Facet	93
Table 107 – Historical Data Insert Client Facet	93
Table 108 – Historical Data Delete Client Facet.....	93
Table 109 – Historical Access Client Server Timestamp Facet	93
Table 110 – Historical Access Modified Data Client Facet.....	94

Table 111 – Historical Structured Data AtTime Client Facet	94
Table 112 – Historical Structured Data Access Client Facet	94
Table 113 – Historical Structured Data Modified Client Facet	94
Table 114 – Historical Structured Data Delete Client Facet	95
Table 115 – Historical Structured Data Update Client Facet	95
Table 116 – Historical Structured Data Replace Client Facet	95
Table 117 – Historical Structured Data Insert Client Facet	95
Table 118 – Historical Events Client Facet	95
Table 119 – Historical Event Update Client Facet	96
Table 120 – Historical Event Replace Client Facet	96
Table 121 – Historical Event Delete Client Facet	96
Table 122 – Historical Event Insert Client Facet	96
Table 123 – Aggregate Subscriber Client Facet	97
Table 124 – User Token – Anonymous Facet	98
Table 125 – User Token – User Name Password Server Facet	98
Table 126 – User Token – X509 Certificate Server Facet	98
Table 127 – User Token – Issued Token Server Facet	98
Table 128 – User Token – Issued Token Windows Server Facet	99
Table 129 – User Token – User Name Password Client Facet	99
Table 130 – User Token – X509 Certificate Client Facet	99
Table 131 – User Token – Issued Token Client Facet	99
Table 132 – User Token – Issued Token Windows Client Facet	99
Table 133 – UA-TCP UA-SC UA Binary	100
Table 134 – SOAP-HTTP WS-SC UA XML	100
Table 135 – SOAP-HTTP WS-SC UA Binary	100
Table 136 – SOAP-HTTP WS-SC UA XML-UA Binary	100
Table 137 – HTTPS UA Binary	101
Table 138 – HTTPS UA XML	101
Table 139 – Security User Access Control Full	101
Table 140 – Security User Access Control Base	101
Table 141 – Security Time Synchronization	102
Table 142 – Best Practice – Audit Events	102
Table 143 – Best Practice – Alarm Handling	102
Table 144 – Best Practice – Random Numbers	102
Table 145 – Best Practice – Timeouts	102
Table 146 – Best Practice – Administrative Access	103
Table 147 – Best Practice – Strict Message Handling	103
Table 148 – Best Practice – Audit Events Client	103
Table 149 – SecurityPolicy – None	103
Table 150 – SecurityPolicy – Basic128Rsa15	104
Table 151 – SecurityPolicy – Basic256	104
Table 152 – SecurityPolicy – Basic256Sha256	105
Table 153 – TransportSecurity – TLS 1.0	105

Table 154 – TransportSecurity – TLS 1.1	105
Table 155 – TransportSecurity – TLS 1.2	106

Withdrawn

INTERNATIONAL ELECTROTECHNICAL COMMISSION

OPC UNIFIED ARCHITECTURE –

Part 7: Profiles

FOREWORD

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

International Standard IEC 62541-7 has been prepared by subcommittee 65E: Devices and integration in enterprise systems, of IEC technical committee 65: Industrial-process measurement, control and automation.

This second edition cancels and replaces the first edition published in 2012. This edition constitutes a technical revision.

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

- a) Added a large number of new Facets to cover additional functional areas of OPC UA. Most significantly:
 - Facets for Historical Access;
 - Facets for Aggregates;
 - Facets for HTTPs

- New Security Facets
- New User Token Facet that supports anonymous access
- Best Practice Facets,

b) New Security Policy for asymmetric key length > 2048

The text of this standard is based on the following documents:

CDV	Report on voting
65E/378/CDV	65E/406/RVC

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

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

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

The committee has decided that the contents of this publication will remain unchanged until the stability date indicated on the IEC web site under "<http://webstore.iec.ch>" in the data related to the specific publication. At this date, the publication will be

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

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

OPC UNIFIED ARCHITECTURE –

Part 7: Profiles

1 Scope

This part of IEC 62541 describes the OPC Unified Architecture (OPC UA) *Profiles*. The *Profiles* in this document are used to segregate features with regard to testing of OPC UA products and the nature of the testing (tool based or lab based). This includes the testing performed by the OPC Foundation provided OPC UA CTT (a self-test tool) and by the OPC Foundation provided Independent certification test labs. This could equally as well refer to test tools provided by another organization or a test lab provided by another organization. What is important is the concept of automated tool based testing versus lab based testing. The scope of this standard includes defining functionality that can only be tested in an a lab and defining the grouping of functionality that is to be used when testing OPC UA products either in a lab or using automated tools. The definition of actual *TestCases* is not within the scope of this document, but the general categories of *TestCases* are within the scope of this document.

Most OPC UA applications will conform to several, but not all of, the *Profiles*.

2 Normative references

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

IEC TR 62541-1, *OPC unified architecture – Part 1: Overview and concepts*

IEC TR 62541-2, *OPC unified architecture – Part 2: Security model*

IEC 62541-3, *OPC unified architecture – Part 3: Address space model*

IEC 62541-4, *OPC unified architecture – Part 4: Services*

IEC 62541-5, *OPC unified architecture – Part 5: Information model*

IEC 62541-6, *OPC unified architecture – Part 6: Mappings*

IEC 62541-8, *OPC unified architecture – Part 8: Data access*

IEC 62541-9, *OPC unified architecture – Part 9: Alarms and conditions*

IEC 62541-11¹, *OPC unified architecture – Part 11: Historical access*

IEC 62541-13¹, *OPC unified architecture – Part 13: Aggregates*

¹ To be published.

3 Terms, definitions, and conventions

3.1 Terms and definitions

For the purposes of this document, the terms and definitions given in IEC TR 62541-1, IEC TR 62541-2, IEC 62541-3, IEC 62541-4, IEC 62541-6, and IEC 62541-8 as well as the following apply. An overview of the terms defined in this standard and their interaction can be viewed in Figure 1.

3.1.1

application

software program that executes or implements some aspect of OPC UA

Note 1 to entry: The application could run on any machine and perform any function. The application could be software or it could be a hardware application, the only requirement is that it implements OPC UA.

3.1.2

ConformanceUnit

specific set of OPC UA features that can be tested as a single entity

Note 1 to entry: A *ConformanceUnit* can cover a group of services, portions of services or information models. For additional detail see Clause 5.

3.1.3

ConformanceGroup

group of *ConformanceUnits* that is given a name

Note 1 to entry: This grouping is only to assist in organizing *ConformanceUnits*. Typical *ConformanceGroups* include groups for each of the service sets in OPC UA and each of the Information Model standards.

3.1.4

Facet

Profile dedicated to a specific feature that a *Server* or *Client* may require

Note 1 to entry: *Facets* are typically combined to form higher-level *Profiles*. The use of the term *Facet* in the title of a *Profile* indicates that the given *Profile* is not a standalone *Profile*.

3.1.5

FullFeatured Profile

Profile that defines all features necessary to build a functional OPC UA *Application*

Note 1 to entry: A *FullFeatured Profile* in particular adds definitions of the transport and security requirements.

3.1.6

ProfileCategory

arranges *Profiles* into application classes, such as *Server* or *Client*

Note 1 to entry: These categories help determine the type of *Application* that a given *Profile* would be used for. For additional details see 4.4.

3.1.7

TestCase

technical description of a set of steps required to test a particular function or information model

Note 1 to entry: *TestCases* provide sufficient details to allow a developer to implement them in code. *TestCases* also provide a detailed summary of the expected result(s) from the execution of the implemented code and any precondition(s) that must be established before the *TestCase* can be executed.

3.1.8

TestLab

facility that is designated to provide testing services

Note 1 to entry: These services include but are not limited to personal that directly perform testing, automated testing and a formal repeatable process. The OPC Foundation has provided detailed standard describing OPC UA TestLabs and the testing they are to provided (see Compliance Part 8 UA *Server*, Compliance Part 9 UA *Client*).

3.2 Abbreviations

DA	Data Access
HA	Historical Access
HMI	Human Machine Interface
NIST	National Institute of Standard and Technology
PKI	Public Key Infrastructure
RSA	Rivest-Shamir-Adleman
UA	Unified Architecture

4 Overview

4.1 General

The OPC Unified architecture multipart standard describes a number of *Services* and a variety of information models. These *Services* and information models can be referred to as features of a *Server* or *Client*. *Servers* and *Clients* need to be able to describe which features they support and wish to have certified. This document provides a grouping of these features. The individual features are grouped into *ConformanceUnits* which are further grouped into *Profiles*. Figure 1 provides an overview of the interactions between *Profiles*, *ConformanceUnits* and *TestCases*. The large arrows indicate the components that are used to construct the parent. For example a *Profile* is constructed from *Profiles* and *ConformanceUnits*. The figure also illustrates a feature of the OPC UA Compliance Test Tool (CTT), in that it will test if a requested *Profile* passes all *ConformanceUnits*. It will also test all other *ConformanceUnits* and report any other *Profiles* that pass conformance testing. The individual *TestCases* are defined in separate documents see Compliance Part 8 UA *Server* and Compliance Part 9 UA *Client*. The *TestCases* are related back to the appropriate *ConformanceUnits* defined in this standard. This relationship is also displayed by the OPC UA Compliance Test Tool.

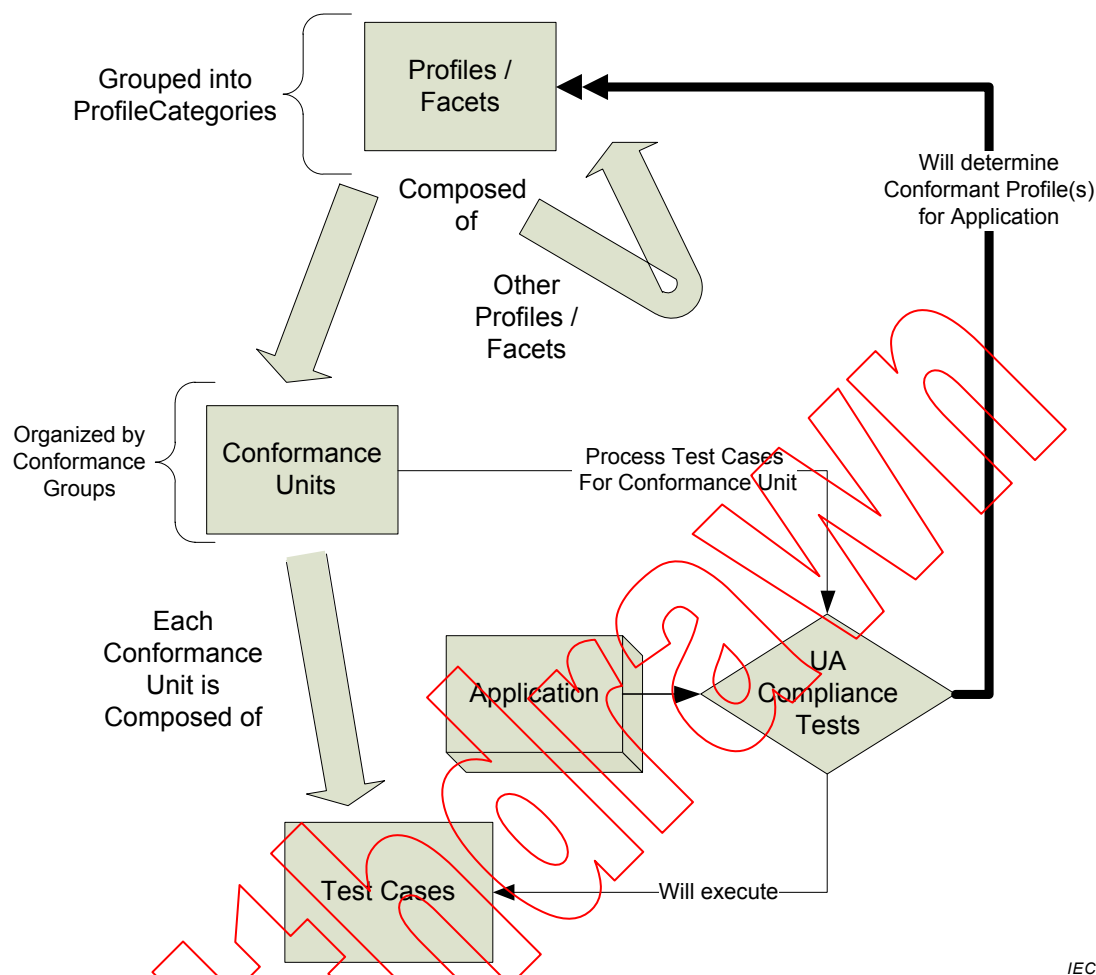


Figure 1 – Profile – ConformanceUnit – TestCases

4.2 ConformanceUnit

Each *ConformanceUnit* represents a specific set of features (e.g. a group of services, portions of services or information models) that can be tested as a single entity. *ConformanceUnits* are the building blocks of a *Profile*. Each *ConformanceUnit* can also be used as a test category. For each *ConformanceUnit*, there would be a number of *TestCases* that test the functionality described by the *ConformanceUnit*. The description of a *ConformanceUnit* is intended to provide enough information to illustrate the required functionality, but in many cases to obtain a complete understanding of the *ConformanceUnit* the reader may be required to also examine the appropriate part of IEC 62541. Additional Information regarding testing of a *ConformanceUnit* are provided in the Compliance Part 8 UA Server or Compliance Part 9 UA Client test standards.

The same features do not appear in more than one *ConformanceUnit*.

4.3 Profiles

A *Profile* is a named aggregation of *ConformanceUnits* and other *Profiles*. To support a *Profile*, an application has to support the *ConformanceUnits* and all aggregated *Profiles*. The definition of *Profiles* is an ongoing activity, in that it is expected that new *Profiles* will be added in the future.

An OPC UA Application will typically support multiple *Profiles*.

Multiple *Profiles* may include the same *ConformanceUnit*.

Testing of a *Profile* consists of testing the individual *ConformanceUnits* that comprise the *Profile*.

Profiles are named based on naming conventions (see 6.3 for details).

4.4 Profile Categories

Profiles are grouped into categories to help vendors and end users understand the applicability of a *Profile*. A *Profile* can be assigned to more than one category.

Table 1 contains the list of currently defined *ProfileCategories*.

Table 1 – ProfileCategories

Category	Description
Client	<i>Profiles</i> of this category specify functions of an OPC UA <i>Client</i> . The URI of such <i>Profiles</i> can be part of a Software <i>Certificate</i> passed in the ActivateSession request.
Security	<i>Profiles</i> of this category specify security related functions. Security policies are part of this category. The URI of security policies has to be part of an Endpoint Description returned from the GetEndpoints service. <i>Profiles</i> of this category apply to <i>Servers</i> and <i>Clients</i> .
Server	<i>Profiles</i> of this category specify functions of an OPC UA <i>Server</i> . The URI of such <i>Profiles</i> can be part of a Software <i>Certificate</i> returned with the CreateSession service response and exposed in the server capabilities.
Transport	<i>Profiles</i> of this category specify specific protocol mappings. The URI of such <i>Profiles</i> has to be part of an Endpoint Description. These <i>Profiles</i> apply to <i>Servers</i> and <i>Clients</i> .

5 ConformanceUnits

5.1 Overview

A *ConformanceUnit* represents an individually testable entity. For improved clarity, the large list of *ConformanceUnits* is arranged into named *ConformanceGroups*. These groups reflect the *Service Sets* in IEC 62541-4 and the OPC UA information models. Table 2 lists the *ConformanceGroups*. These groups and the *ConformanceUnits* that they describe are detailed in the Subclauses of Clause 5 starting with 5.2 *ConformanceGroups* have no impact on testing; they are used only for organizational reasons, i.e. to simplify the readability of this document.

Table 2 – ConformanceGroups

Group	Description
Address Space Model	Defines <i>ConformanceUnits</i> for various features of the OPC UA <i>AddressSpace</i> .
Aggregates	All <i>ConformanceUnits</i> that are related to <i>Aggregates</i> , including individual <i>ConformanceUnits</i> for each supported <i>Aggregate</i> as described in IEC 62541-13.
Alarms and Conditions	All <i>ConformanceUnits</i> that are associated with the OPC UA information model for <i>Conditions</i> , acknowledgeable <i>Conditions</i> , confirmations and <i>Alarms</i> as specified in IEC 62541-9.
Attribute Services	Includes <i>ConformanceUnits</i> to read or write current or historical <i>Attribute</i> values.
Auditing	User level security includes support for security audit trails, with traceability between <i>Client</i> and <i>Server</i> audit logs.
Base Information	All information elements as defined in IEC 62541-5.
Data Access	<i>ConformanceUnits</i> specific to <i>Clients</i> and <i>Servers</i> that deal with the representation and use of automation data as specified in IEC 62541-8.
Discovery Services	<i>ConformanceUnits</i> which focus on <i>Server Endpoint Discovery</i> .
Historical Access	Access to archived data of node <i>Attribute</i> values or <i>Events</i> .
Method Services	<i>Methods</i> represent the function calls of <i>Objects</i> . <i>Methods</i> are invoked and return only after completion (successful or unsuccessful).
Miscellaneous	This group contains <i>ConformanceUnits</i> that cover miscellaneous subjects, such as recommended behaviours, documentation etc. These <i>ConformanceUnits</i> typically do not fit into any of the other groups.
Monitored Item Services	<i>Clients</i> define <i>MonitoredItems</i> to subscribe to data and <i>Events</i> . Each <i>MonitoredItem</i> identifies the item to be monitored and the <i>Subscription</i> to use to send <i>Notifications</i> .
Node Management Services	Bundles <i>ConformanceUnits</i> for all <i>Services</i> to add and delete OPC UA <i>AddressSpace Nodes</i> and <i>References</i> .
Protocol and Encoding	Covers all transport and encoding combinations that are specified in IEC 62541-6.
Query Services	A <i>Query</i> may be used to provide advanced filtering and return a subset of data.
Redundancy	The design of OPC UA ensures that vendors can create redundant <i>Clients</i> and redundant <i>Servers</i> in a consistent manner. Redundancy may be used for high availability, fault tolerance and load balancing.
Security	Security related <i>ConformanceUnits</i> that can be profiled this includes all aspects of security.
Session Services	An (OPC UA) <i>Session</i> is an application layer connection.
Subscription Services	<i>Subscriptions</i> are used to report <i>Notifications</i> to the <i>Client</i> .
View Services	<i>Clients</i> use the <i>View Service Set</i> to navigate through the OPC UA <i>AddressSpace</i> or through a <i>View</i> (a subset) of the OPC UA <i>AddressSpace</i> .

5.2 Services

Tables 3 to 10 describe *ConformanceUnits* for the *Services* specified in IEC 62541-4. The tables correlate with the *Service Sets*.

A single *ConformanceUnit* can reference several *Services* (e.g. *CreateSession*, *ActivateSession* and *CloseSession*) but can also refer to individual aspects of *Services* (e.g. the use of *ActivateSession* to impersonate a new user).

Each table includes a listing of the *Profile Category* to which a *ConformanceUnit* belongs, the title and description of the *ConformanceUnit* and a column that indicates if the *ConformanceUnit* is derived from another *ConformanceUnit*. A *ConformanceUnit* that is derived from another *ConformanceUnit* includes all of the same tests as its parent plus one or more additional TestCases. These TestCases can only further restrict the existing TestCases. An example would be one in which the number of connections is tested, where the TestCase of the parent required at least one connection and the derived *ConformanceUnit* would require a *TestCase* for at least five connections.

The *Discovery Service Set* is composed of multiple *ConformanceUnits* (see Table 3). All *Servers* provide some aspects of this functionality; see *Profiles* categorized as *Server Profiles* for details. *Clients* may support some aspects of this functionality; see *Profiles* categorized as *Client Profiles* for details.

Table 3 – Discovery Services

Category	Title	Description	Derived
Server	Discovery Get Endpoints	Support the GetEndpoints <i>Service</i> to obtain all Endpoints of the <i>Server</i> . This includes filtering based on <i>Profiles</i> .	
Server	Discovery Find Servers Self	Support the FindServers <i>Service</i> only for itself.	
Server	Discovery Register	Call the RegisterServer <i>Service</i> to register itself (OPC UA <i>Server</i>) with an external <i>Discovery Service</i> via a secure channel with a SecurityMode other than "None".	
Server	Discovery Configuration	Allow configuration of the <i>Discovery Server</i> URL where the <i>Server</i> will register itself. Allow complete disabling of registration with a <i>Discovery Server</i> .	
Client	Discovery Client Find Servers Basic	Uses the FindServers <i>Service</i> to obtain all Servers installed on a given platform.	
Client	Discovery Client Find Servers with URI	Use FindServers <i>Service</i> to obtain URLs for specific <i>Server</i> URIs.	
Client	Discovery Client Find Servers Dynamic	Detect new Servers after an initial FindServers <i>Service</i> call.	
Client	Discovery Client Get Endpoints Basic	Uses the GetEndpoints <i>Service</i> to obtain all Endpoints for a given <i>Server</i> URI.	
Client	Discovery Client Get Endpoints Dynamic	Detect changes to the Endpoints after an initial GetEndpoints <i>Service</i> call.	
Client	Discovery Client Configure Endpoint	Allow specification of an Endpoint without going through the <i>Discovery Service Set</i> .	

The *Session Service Set* is composed of multiple *ConformanceUnits* (see Table 4). The CreateSession, ActivateSession, and CloseSession services are supported as a single unit. All *Servers* and *Clients* provide this functionality.

Table 4 – Session Services

Category	Title	Description	Derived
Server	Session General Service Behaviour	Implement basic <i>Service</i> behaviour. This includes in particular: – checking the authentication token – returning the requestHandle in responses – returning available diagnostic information as requested with the 'returnDiagnostics' parameter – respecting a timeoutHint	
Server	Session Base	Support the <i>Session Service Set</i> (CreateSession, ActivateSession, CloseSession) except the use of ActivateSession to change the <i>Session</i> user. This includes correct handling of all parameters that are provided. Note that for the CreateSession and ActivateSession services, if the SecurityMode = None then: 1) The Application <i>Certificate</i> and Nonce are optional. 2) The signatures are null/empty. The details of this are described in IEC 62541-4.	
Server	Session Change User	Support the use of ActivateSession to change the <i>Session</i> user.	
Server	Session Cancel	Support the Cancel <i>Service</i> to cancel outstanding requests.	
Server	Session Minimum 1	Support minimum 1 <i>Session</i> (total).	
Server	Session Minimum 2 Parallel	Support minimum 2 parallel <i>Sessions</i> (total for all <i>Clients</i>).	
Server	Session Minimum 50 Parallel	Support minimum 50 parallel <i>Sessions</i> (total for all <i>Clients</i>).	
Client	Session Client General Service Behaviour	Implement basic <i>Service</i> behaviour. This includes in particular: – including the proper authentication token of the <i>Session</i> – creating a requestHandle if needed – requesting diagnostic information with the 'returnDiagnostics' parameter – evaluate the serviceResult and operational results	
Client	Session Client Base	Use the <i>Session Service Set</i> (CreateSession, ActivateSession, and CloseSession) except the use of ActivateSession to change the <i>Session</i> user. This includes correct handling of all parameters that are provided. Note that for the CreateSession and ActivateSession services, if the SecurityMode = None then: 1) The Application <i>Certificate</i> and Nonce are optional. 2) The signatures are null/empty.	
Client	Session Client Multiple Connections	Support unlimited connections (client side) with multiple <i>Servers</i> . Any limit on numbers of connections is from server side. May have a memory based limit, but not a software constraint limit.	

Category	Title	Description	Derived
Client	Session Client Renew NodeIds	This <i>ConformanceUnit</i> applies to <i>Clients</i> that allow persisting NodeIds. Verify that the Namespace Table has not changed for NodeIds that the <i>Client</i> has persisted and is going to re-use beyond a <i>Session</i> lifetime. If changes occurred the <i>Client</i> has to recalculate the Namespace Indices of the respective NodeIds.	
Client	Session Client Impersonate	Uses ActivateSession to change the <i>Session</i> user (impersonation).	
Client	Session Client KeepAlive	Make periodic requests to keep the <i>Session</i> alive.	
Client	Session Client Detect Shutdown	Read or monitor the ServerStatus/State <i>Variable</i> to recognize a potential shutdown of the <i>Server</i> and clean up resources.	
Client	Session Client Cancel	Use the Cancel <i>Service</i> to cancel outstanding requests.	
Client	Session Client Auto Reconnect	Automatic <i>Client</i> reconnect including: – ActivateSession with new SecureChannel if SecureChannel is no longer valid but <i>Session</i> is still valid – Creation of a new <i>Session</i> only if <i>Session</i> is no longer valid	
Client	Client Entry-Level Support	The <i>Client</i> is able to interoperate with <i>Servers</i> with lowest level functionality. This includes the ability to operate with a single <i>Session</i> , a pre-knowledge of the OPC UA Types (the <i>Server</i> may not expose them in the <i>AddressSpace</i>), and the ability to use Read vs. <i>Subscriptions</i> for monitoring. There may be further restrictions provided by the <i>Server</i> via the <i>Server</i> capabilities.	

The *Node Management Service Set* is composed of multiple *ConformanceUnits* (see Table 5). *Servers* may provide some aspects of this functionality; see *Profiles* categorized as *Server Profiles* for details. *Clients* may support some aspects of this functionality; see *Profiles* categorized as *Client Profiles* for details.

Table 5 – Node Management Services

Category	Title	Description	Derived
Server	Node Management Add Node	Support the AddNodes <i>Service</i> to add one or more <i>Nodes</i> into the OPC UA <i>AddressSpace</i> .	
Server	Node Management Delete Node	Support the DeleteNodes <i>Service</i> to delete one or more <i>Nodes</i> from the OPC UA <i>AddressSpace</i> .	
Server	Node Management Add Ref	Support the AddReferences <i>Service</i> to add one or more <i>References</i> to one or more <i>Nodes</i> in the OPC UA <i>AddressSpace</i> .	
Server	Node Management Delete Ref	Support the DeleteReferences <i>Service</i> to delete one or more <i>References</i> of a <i>Node</i> in the OPC UA <i>AddressSpace</i> .	
Client	Node Management Client	Uses <i>Node Management Services</i> to add or delete <i>Nodes</i> and to add or delete <i>References</i> in <i>Server's</i> OPC UA <i>AddressSpace</i> .	

The View *Service* Set is composed of a multiple *ConformanceUnits* (see Table 6). All *Servers* support some aspects of this conformance group. *Clients* may support some aspects of this functionality; see *Profiles* categorized as *Client Profiles* for details.

Table 6 – View Services

Category	Title	Description	Derived
Server	View Basic	Support the View <i>Service</i> Set (Browse, BrowseNext).	
Server	View TranslateBrowsePath	Support TranslateBrowsePathsToNodeIds <i>Service</i> .	
Server	View RegisterNodes	Support the RegisterNodes and UnregisterNodes <i>Services</i> as a way to optimize access to repeatedly used <i>Nodes</i> in the <i>Server's</i> OPC UA <i>AddressSpace</i> .	
Server	View Minimum Continuation Point 01	Support minimum 1 continuation point per <i>Session</i> .	
Server	View Minimum Continuation Point 05	Support minimum 5 continuation points per <i>Session</i> . This number has to be supported for at least half of the minimum required sessions.	
Client	View Client Basic Browse	Uses Browse and BrowseNext <i>Services</i> to navigate through the <i>Server's</i> OPC UA <i>AddressSpace</i> . Make use of the referenceTypeId and the nodeClassMask to specify the needed <i>References</i> .	
Client	View Client Basic ResultSet Filtering	Makes use of the resultMask parameter to optimize the result set to be returned by the <i>Server</i> .	
Client	View Client TranslateBrowsePath	Uses the TranslateBrowsePathsToNodeIds <i>Service</i> to identify the NodeIds for <i>Nodes</i> where a starting <i>Node</i> and a BrowsePath is known. Makes use of bulk operations rather than multiple calls whenever possible.	
Client	View Client RegisterNodes	Uses the RegisterNodes <i>Service</i> to optimize access for <i>Nodes</i> that are used repeatedly. Use UnregisterNodes when <i>Nodes</i> are not used anymore.	

The *Attribute Service* Set is composed of multiple *ConformanceUnits* (see Table 7). The majority of the *Attribute* service set is a core functionality of OPC UA and as such is supported by most *Servers*. Most *Clients* will also support some aspects of the *Attribute Service* Set

Table 7 – Attribute Services

Category	Title	Description	Derived
Server	Attribute Read	Supports the Read <i>Service</i> to read one or more <i>Attributes</i> of one or more <i>Nodes</i> . This includes support of the <i>IndexRange</i> parameter to read a single element or a range of elements when the <i>Attribute</i> value is an array.	
Server	Attribute Read Complex	Supports reading and encoding Values with Structured DataTypes.	
Server	Attribute Write Values	Supports writing to values to one or more <i>Attributes</i> of one or more <i>Nodes</i> .	
Server	Attribute Write Complex	Supports writing and decoding Values with Structured DataTypes.	
Server	Attribute Write StatusCode & Timestamp	Supports writing of <i>StatusCode</i> and <i>Timestamps</i> along with the Value.	
Server	Attribute Write Index	Supports the <i>IndexRange</i> to write a single element or a range of elements when the <i>Attribute</i> value is an array.	
Server	Attribute Alternate Encoding	Supports alternate Data Encoding when reading value <i>Attributes</i> . By default, every <i>Server</i> has to support the Data Encoding of the currently used <i>Stack Profile</i> (i.e. binary with UA Binary Encoding and XML with XML Encoding). This <i>ConformanceUnit</i> – when supported – specifies that the other Data Encoding is supported in addition.	
Server	Attribute Historical Read	Supports the <i>HistoryRead Service</i> . The details of what aspects of this service are used are listed in additional <i>ConformanceUnits</i> , but at least one of <i>ReadRaw</i> , <i>ReadProcessed</i> , <i>ReadModified</i> , <i>ReadAtTime</i> or <i>ReadEvents</i> must be supported.	
Server	Attribute Historical Update	Supports the <i>HistoryUpdate service</i> . The details of the supported features of this service are described by additional <i>ConformanceUnits</i> , but at least one of the following must be supported: <i>InsertData</i> , <i>InsertEvents</i> , <i>ReplaceData</i> , <i>ReplaceEvents</i> , <i>UpdateData</i> , <i>UpdateEvents</i> , <i>DeleteData</i> , <i>DeleteEvents</i> or <i>DeleteAtTime</i> .	
Client	Attribute Client Read Base	Use the Read <i>Service</i> to read one or more <i>Attributes</i> of one or more <i>Nodes</i> . This includes use of an <i>IndexRange</i> to select a single element or a range of elements when the <i>Attribute</i> value is an array. <i>Clients</i> shall use bulk operations whenever possible to reduce the number of <i>Service</i> invocations.	
Client	Attribute Client Read with proper Encoding	This <i>ConformanceUnit</i> refers to the ability of a <i>Server</i> to support more than one Data Encoding for <i>Attribute</i> values. <i>Clients</i> can discover the available encodings and can explicitly choose one when calling the Read <i>Service</i> .	
Client	Attribute Client Read Complex	Read and decode Values with Structured DataTypes.	

Category	Title	Description	Derived
Client	Attribute Client Write Base	Use the Write <i>Service</i> to write values to one or more <i>Attributes</i> of one or more <i>Nodes</i> . This includes use of an <i>IndexRange</i> to select a single element or a range of elements when the <i>Attribute</i> value is an array. <i>Clients</i> shall use bulk operations whenever possible to reduce the number of <i>Service</i> invocations.	
Client	Attribute Client Write Complex	Write and Encode Values with Structured <i>DataTypes</i> .	
Client	Attribute Client Write Quality & TimeStamp	Use the Write <i>Service</i> to also write <i>StatusCode</i> and/or <i>Timestamps</i> along with a <i>Value</i> .	
Client	Attribute Client Historical Read	The <i>Client</i> makes use of the <i>HistoryRead</i> service. The details of which aspect of this service are used are provided by additional <i>ConformanceUnits</i> , but at least one or more of the following is used: <i>ReadRaw</i> , <i>ReadAtTime</i> , <i>ReadProcessed</i> , <i>ReadModified</i> or <i>ReadEvents</i> .	
Client	Attribute Client Historical Updates	The <i>Client</i> makes use of the <i>HistoryUpdate</i> service. The details of this usage are provided by additional <i>ConformanceUnits</i> , but at least one or more of the following must be provided: <i>InsertData</i> , <i>InsertEvent</i> , <i>ReplaceData</i> , <i>ReplaceEvent</i> , <i>UpdateData</i> , <i>UpdateEvents</i> , <i>DeleteData</i> or <i>DeleteEvents</i> or <i>DeleteAtTime</i> .	

The *Method Service Set* is composed of *ConformanceUnits* (see Table 8). The primary *ConformanceUnits* provide support for the call functionality. *Servers* may provide some aspects of this functionality; see *Profiles* categorized as *Server Profiles* for details. *Clients* may support some aspects of this functionality; see *Profiles* categorized as *Client Profiles* for details.

Table 8 – Method Services

Category	Title	Description	Derived
Server	Method Call	Support the Call <i>Service</i> to call (invoke) a <i>Method</i> which includes support for <i>Method Parameters</i> .	
Client	Method Client Call	Use the Call <i>Service</i> to call one or several <i>Methods</i> .	

The *MonitoredItem Service Set* is composed of multiple *ConformanceUnits* (see Table 9). *Servers* may provide some aspects of this functionality; see *Profiles* categorized as *Server Profiles* for details. *Clients* may support some aspects of this functionality; see *Profiles* categorized as *Client Profiles* for details.

Table 9 – Monitored Item Services

Category	Title	Description	Derived
Server	Monitor Basic	Support the following <i>MonitoredItem</i> Services: CreateMonitoredItems, ModifyMonitoredItems, DeleteMonitoredItems and SetMonitoringMode.	
Server	Monitor Value Change	Support creation of <i>MonitoredItems</i> for <i>Attribute</i> value changes. This includes support of the <i>IndexRange</i> to select a single element or a range of elements when the <i>Attribute</i> value is an array.	
Server	Monitored Items Deadband Filter	Supports an absolute Deadband filter as a <i>DataChangeFilter</i> for numeric data types.	
Server	Monitor Aggregate Filter	Support for Aggregate filters for <i>MonitoredItems</i> . The result of this <i>ConformanceUnit</i> includes a list of Aggregates that are supported as part of the <i>Profile Certificate</i> .	
Server	Monitor Alternate Encoding	Support alternate encoding when monitoring value <i>Attributes</i> . By default, every <i>Server</i> has to support the encoding of the currently used <i>Stack Profile</i> (i.e. binary with UA Binary Encoding and XML with XML Encoding). This <i>ConformanceUnit</i> – when supported – specifies that the other encoding is supported in addition.	
Server	Monitor Items 2	Support at least 2 <i>MonitoredItems</i> per <i>Subscription</i> .	
Server	Monitor Items 10	Support at least 10 <i>MonitoredItems</i> per <i>Subscription</i> .	
Server	Monitor Items 100	Support at least 100 <i>MonitoredItems</i> per <i>Subscription</i> . This number has to be supported for at least half of the required <i>Subscriptions</i> for half of the required <i>Sessions</i> .	
Server	Monitor Items 500	Support at least 500 <i>MonitoredItems</i> per <i>Subscription</i> . This number has to be supported for at least half of the required <i>Subscriptions</i> for half of the required <i>Sessions</i> .	
Server	Monitor QueueSize_1	This <i>ConformanceUnit</i> does not require queuing when multiple value changes occur during a “publish period”. I.e. the latest change will be sent in the <i>Notification</i> .	
Server	Monitor MinQueueSize_02	Support at least 2 queue entries for <i>MonitoredItems</i> . <i>Servers</i> often will adapt the queue size to the number of currently <i>MonitoredItems</i> . However, it is expected that <i>Servers</i> support this minimum queue size for at least one third of the supported <i>MonitoredItems</i> .	

Category	Title	Description	Derived
Server	Monitor MinQueueSize_05	Support at least 5 queue entries for <i>MonitoredItems</i> . <i>Servers</i> often will adapt the queue size to the number of currently <i>MonitoredItems</i> . However, it is expected that <i>Servers</i> support this minimum queue size for at least one third of the supported <i>MonitoredItems</i> .	
Server	Monitor QueueSize_ServerMax	This <i>ConformanceUnit</i> is for events. When the Client requests queueSize=MAXUInt32 the <i>Server</i> is to return the maximum queue size that it can support for event notifications as the revisedQueueSize.	
Server	Monitor Triggering	Support the SetTriggering <i>Service</i> to create and/or delete triggering links for a triggering item.	
Server	Monitor Events	Support creation of <i>MonitoredItems</i> for an "EventNotifier Attribute" for the purpose of <i>Event Notification</i> . The subscription includes supporting a filter that includes SimpleAttribute Operands and a select list of Operators. The list of Operators includes: Equals, IsNull, GreaterThan, LessThan, GreaterThanorEqual, LessThanorEqual, Like, Not, Between, InList, And, Or, Cast, BitwiseAnd, BitwiseOr.	
Server	Monitor Complex Event Filter	Support for complex <i>Event</i> filters, where complex is defined as supporting the complex filter operator (TypeOf).	
Client	Monitor Client Value Change	Use the <i>MonitoredItem Service Set</i> to register items for changes in <i>Attribute</i> value. Use CreateMonitoredItems to register the <i>Node/Attribute</i> tuple. Set proper sampling interval, Deadband filter and queuing mode. Use disabling / enabling instead of deleting and re-creating a <i>MonitoredItem</i> . Use bulk operations rather than individual service requests to reduce communication overhead.	
Client	Monitor Client Deadband Filter	Uses Absolute Deadband filters for subscriptions.	
Client	Monitor Client by Index	Use the IndexRange to select a single element or a range of elements when the <i>Attribute</i> value is an array.	
Client	Monitor Client Aggregate Filter	Uses Aggregate filters for Subscriptions.	
Client	Monitor Client Events	Use the <i>MonitoredItem Service Set</i> to create <i>MonitoredItems</i> for <i>Event</i> notifications.	
Client	Monitor Client Event Filter	Use the <i>Event</i> filter when calling CreateMonitoredItems to filter the desired Events and to select the columns to be provided for each <i>Event Notification</i> .	

Category	Title	Description	Derived
Client	Monitor Client Complex Event Filter	Uses complex <i>Event</i> filters.	
Client	Monitor Client Modify	Use <i>ModifyMonitoredItems Service</i> to change the configuration setting. Use <i>SetMonitoringMode Service</i> to disable / enable sampling and / or publishing.	
Client	Monitor Client Trigger	Use the Triggering Model if certain items are to be reported only if some other item triggers. Use proper monitoring mode for these items. Use <i>SetTriggering Service</i> to link these items to the trigger item.	

The *Subscription Service Set* is composed of multiple *ConformanceUnits* (see Table 10). *Servers* may provide some aspects of this functionality; see *Profiles* categorized as *Server Profiles* for details. *Clients* may support some aspects of this functionality; see *Profiles* categorized as *Client Profiles* for details.

Withdrawal

Table 10 – Subscription Services

Category	Title	Description	Derived
Server	Subscription Basic	Support the following <i>Subscription Services</i> : CreateSubscription, ModifySubscription, DeleteSubscriptions, Publish, Republish and SetPublishingMode.	
Server	Subscription Minimum 1	Support at least 1 Subscriptions per <i>Session</i> . This number has to be supported for all of the minimum required sessions.	
Server	Subscription Minimum 02	Support at least 2 Subscriptions per <i>Session</i> . This number has to be supported for at least half of the minimum required sessions.	
Server	Subscription Minimum 05	Support at least 5 Subscriptions per <i>Session</i> . This number has to be supported for at least half of the minimum required sessions.	
Server	Subscription Publish Min 02	Support at least 2 Publish <i>Service</i> requests per <i>Session</i> . This number has to be supported for all of the minimum required sessions. Support of republish is optional and no notification retransmission queue has to be provided however the republish service must be provided and will return the appropriate operation level results.	
Server	Subscription Publish Min 05	Support at least 5 Publish <i>Service</i> requests per <i>Session</i> . This number has to be supported for at least half of the minimum required sessions. Support, as a minimum, the number of Publish requests per session as the size of the NotificationMessage retransmission queue for Republish.	
Server	Subscription Publish Min 10	Support at least 10 Publish <i>Service</i> requests per <i>Session</i> . This number has to be supported for at least half of the minimum required sessions. Support as a minimum, the number of Publish requests per session as the size of the NotificationMessage retransmission queue for Republish.	
Server	Subscription Publish Discard Policy	Respect the specified policy for discarding Publish <i>Service</i> requests. If the maximum number of Publish <i>Service</i> requests has been queued and a new Publish <i>Service</i> request arrives, the "oldest" Publish request has to be discarded by returning the proper error.	
Server	Subscription Transfer	Support TransferSubscriptions <i>Service</i> to transfer a <i>Subscription</i> from one <i>Session</i> to another.	

Category	Title	Description	Derived
Client	Subscription Basic	Use the <i>Subscription</i> and <i>MonitoredItem Service Set</i> as an efficient means to detect changes of <i>Attribute</i> values and / or to receive <i>Event</i> occurrences. Set appropriate intervals for publishing, keep alive notifications and total <i>Subscription</i> lifetime. Supply a sufficient number of Publish requests to the <i>Server</i> so that <i>Notifications</i> can be sent whenever a publish timer expires. Acknowledge received <i>Notifications</i> with subsequent Publish requests.	
Client	Subscription Republish	Evaluate the sequence number in <i>Notifications</i> to detect lost <i>Notifications</i> . Use Republish to request missing <i>Notifications</i> .	
Client	Subscription Modify	Allow modification of the <i>Subscription</i> configuration using the <i>ModifySubscription Service</i> .	
Client	Subscription Transfer	The <i>Client</i> supports transferring <i>Subscription</i> from other <i>Clients</i> . This <i>ConformanceUnit</i> is used as part of redundant <i>Clients</i> .	
Client	Subscription Multiple	Use multiple Subscriptions to reduce the payload of individual <i>Notifications</i> .	
Client	Subscription Publish Configurable	Send multiple Publish <i>Service</i> requests to assure that the <i>Server</i> is always able to send <i>Notifications</i> . The number of parallel Publish <i>Service</i> requests per <i>Session</i> shall be configurable.	

5.3 Transport and communication related features

Table 11 describes security related *ConformanceUnits*. All of these *ConformanceUnits* apply equally to both *Clients* and *Servers*, where a *Client* uses the related security unit and a *Server* supports the use of it. These items are defined in detail in IEC 62541-6. It is recommended that a *Server* and *Client* support as many of these options as possible in order to achieve increased levels of interoperability. It is the task of an administrator to determine which of these *ConformanceUnits* are exposed in a given deployed *Server* or *Client* application.

Table 11 – Security

Category	Title	Description	Derived
Security	Security Certificate Validation	A certificate will be validated as specified in IEC 62541-4. This includes among others structure and signature examination. Allowing for some validation errors to be suppressed by administration directive.	
Security	Security None	A suite of algorithms that does NOT provide any security settings: -> SymmetricSignatureAlgorithm – Not Used -> SymmetricEncryptionAlgorithm – Not Used -> AsymmetricSignatureAlgorithm – Not Used -> SymmetricKeyWrapAlgorithm – Not Used -> AsymmetricEncryptionAlgorithm – Not Used -> KeyDerivationAlgorithm – Not Used -> DerivedSignatureKeyLength – 0 The use of this suite of algorithms must be able to be enabled or disabled by an administrator.	
Security	Security CreateSession ActivateSession	None When SecurityPolicy=None, the CreateSession and ActivateSession service allow for a NULL/empty signature and do not require Application Certificates or a Nonce.	

Category	Title	Description	Derived
Security	Security Basic 128Rsa15	A suite of algorithms that uses RSA15 as Key-Wrap-algorithm and 128-Bit for encryption algorithms. -> SymmetricSignatureAlgorithm – HmacSha1 – (http://www.w3.org/2000/09/xmlsig#hmac-sha1). -> SymmetricEncryptionAlgorithm – Aes128 – (http://www.w3.org/2001/04/xmlenc#aes128-cbc). -> AsymmetricSignatureAlgorithm – RsaSha1 – (http://www.w3.org/2000/09/xmlsig#rsa-sha1). -> AsymmetricKeyWrapAlgorithm – KwRsa15 – (http://www.w3.org/2001/04/xmlenc#rsa-1_5). -> AsymmetricEncryptionAlgorithm – Rsa15 – (http://www.w3.org/2001/04/xmlenc#rsa-1_5). -> KeyDerivationAlgorithm – PSha1 – (http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/dk/p_sha1). -> DerivedSignatureKeyLength – 128. -> MinAsymmetricKeyLength – 1024 -> MaxAsymmetricKeyLength – 2048 -> CertificateSignatureAlgorithm – Sha1	

Category	Title	Description	Derived
Security	Security Basic 256	A suite of algorithms that are for 256-Bit encryption, algorithms include: -> SymmetricSignatureAlgorithm – HmacSha1 (http://www.w3.org/2000/09/xmlsig#hmac-sha1). -> SymmetricEncryptionAlgorithm – Aes256 (http://www.w3.org/2001/04/xmlenc#aes256-cbc). -> AsymmetricSignatureAlgorithm – RsaSha1 (http://www.w3.org/2000/09/xmlsig#rsa-sha1). -> AsymmetricKeyWrapAlgorithm – KwRsaOaep (http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p). -> AsymmetricEncryptionAlgorithm – RsaOaep (http://www.w3.org/2001/04/xmlenc#rsa-oaep). -> KeyDerivationAlgorithm – PSha1 – (http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/dk/p_sha1) -> DerivedSignatureKeyLength – 192. -> MinAsymmetricKeyLength – 1024 -> MaxAsymmetricKeyLength – 2048 -> CertificateSignatureAlgorithm – Sha1	

Category	Title	Description	Derived
Security	Security Basic 256 Sha256	A suite of algorithms that are for 256-Bit encryption, algorithms include. -> SymmetricSignatureAlgorithm – Hmac_Sha256 (http://www.w3.org/2000/09/xmlsig#hmac-sha256). -> SymmetricEncryptionAlgorithm – Aes256_CBC (http://www.w3.org/2001/04/xmlenc#aes256-cbc). -> AsymmetricSignatureAlgorithm – Rsa_Sha256 (http://www.w3.org/2000/09/xmlsig#rsa-sha256). -> AsymmetricKeyWrapAlgorithm – KwRsaOaep (http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p). -> AsymmetricEncryptionAlgorithm – Rsa_Oaep (http://www.w3.org/2001/04/xmlenc#rsa-oaep). -> KeyDerivationAlgorithm – PSHA256 (http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/dk/p_sha256). -> DerivedSignatureKeyLength – 256 -> MinAsymmetricKeyLength – 2048 -> MaxAsymmetricKeyLength – 4096 -> CertificateSignatureAlgorithm – Sha256 Support for this security profile may require support for a second application instance certificate, with a larger keysize. Applications shall support multiple Application Instance <i>Certificates</i> if required by supported Security Policies and use the certificate that is required for a given security endpoint.	
Security	Security TLS General	This <i>ConformanceUnit</i> indicates that at least one of the transport security <i>Profiles</i> for TLS is supported by this application. It is used in TLS transport <i>Profiles</i>, but the choice of transport security profile is optional. The actual used security profile will default to the most secure one.	
Security	Security TLS 1.1	The connection is established using TLS 1.1. The application needs to be configured to prevent TLS 1.0 connections, unless the TLS 1.0 connection is using TLS_RSA_WITH_RC4_128_SHA as described in <i>ConformanceUnit</i> "Security TLS_RSA_WITH_RC4_128_SHA"	

Category	Title	Description	Derived
Security	Security TLS_RSA_WITH_RC4_128_SHA	The connection is established using TLS_RSA_WITH_RC4_128_SHA. The application needs to be configured to prevent the use of AES based protocol suites (TLS 1.0).	
Security	Security TLS_RSA_WITH_AES_256_CBC_SHA256	The connection is established using TLS_RSA_WITH_AES_256_CBC_SHA256. That has a MinAsymmetricKeyLength - 2048, MaxAsymmetricKeyLength - 4096, AsymmetricSignatureAlgorithm - RSA_SHA256. (TLS 1.2)	
Security	Security Encryption Required	Encryption is required using the algorithms provide in the security algorithm suite.	
Security	Security Signing Required	Signing is required using the algorithms provide in the security algorithm suite.	
Security	Security Time Synch - Configuration	Application supports configuring acceptable clock skew.	
Security	Security Time Synch - NTP / OS Based support	Application supports time synchronization, either via an implementation of Network Time Protocol (NTP), or via features of a standard operating system.	
Security	Security Time Synch - UA based support	An application makes use of the responses header timestamp provided by a configured well know source, such as a <i>Discovery Server</i> to synchronize the time on the application and that this time synchronization occurs periodically. Use of this TimeSyncing can be configured.	
Security	Security Administration	Allow configuration of the following Security related items. * select the allowed User identification policy or policies (User Name/Password or X509 or Kerberos or Anonymous). * enable/disable the security policy "None" or other security policies. * enable/disable endpoints with MessageSecurityMode SIGN or SIGNANDENCRYPT. * set the permitted certification authorities. * define how to react to unknown <i>Certificates</i> .	
Security	Security Administration - XML Schema	Support the OPC UA defined XML schema for importing and exporting security configuration information. This schema is defined in IEC 62541-6.	
Security	Security Administration Certificate	Allow a site administrator to be able to assign a site specific ApplicationInstanceCertificate and if desired to configure a site specific <i>Certificate Authority</i> (CA).	

Category	Title	Description	Derived
Security	Security Default ApplicationInstanceCertificate	An application, when installed, has a default ApplicationInstanceCertificate that is valid. The default ApplicationInstanceCertificate shall either be created as part of the installation or installation instructions explicitly describe the process to create and apply a default ApplicationInstanceCertificate to the application.	
Security	Security – No Application Authentication	The Server supports being able to be configured for no application authentication, just User authentication and normal encryption/signing: - Configure server to accept all certificates - Certificates are just used for message security (signing and encryption) - Users level is used for authentication	
Security	Best Practice – Audit Events	Subscriptions for Audit Events are restricted to authorized personnel. A Server may also reject a Subscription for Audit Events that is not over a Secure Channel if one is available.	
Security	Best Practice – Alarm Handling	A Server should restrict critical alarm functionality to users that have the appropriate rights to perform these actions. This would include disabling or alarms, shelving of alarms and generation of dialog messages. It would also include other security related functionality such maintaining appropriate timeouts for shelving and dialogs and preventing an overload of dialog messages.	
Security	Best Practice – Random Numbers	All random numbers that are required for security use appropriate cryptographic library based random number generators.	
Security	Best Practice – Timeouts	The user is able to configure reasonable timeouts for Secure Channels, Sessions and Subscriptions to limit denial of service and resource consumption issues (see IEC TR 62541-2 for additional details).	
Security	Best Practice – Administrative Access	The Server and Client allow for appropriate restriction of access to administrative personnel. This includes multiple levels of administrative access on platforms that support multiple administrative roles (such as Windows or Linux).	
Security	Best Practice – Strict Message Handling	The application assures that messages that are illegally or incorrectly formed are rejected with appropriate error codes or appropriate actions as specified in IEC 62541-4 and IEC 62541-6.	

Category	Title	Description	Derived
Security	Best Practice – Audit Events Client	Audit tracking system connects to a Server using a Secure Channel and under the appropriate administrative rights to allow access to Audit Events.	
Security	Security User Name Password	The Server supports User Name/Password combination(s). Encryption of the password with the algorithm provided in the UserNameIdentityToken is required if no message encryption is used.	
Security	Security User X509	The Server supports a public/private key pair for user identity. The use of this feature must be able to be enabled or disabled by an administrator.	
Security	Security User IssuedToken Kerberos	The Server supports a Kerberos Server token for User Identity. The use of this feature must be able to be enabled or disabled by an Administrator. Specific encryption of the IssuedToken is required if no message encryption is used. The use of this token is defined in Kerberos Token Documentation.	
Security	Security User IssuedToken Kerberos Windows	The Server supports the Windows implementation of Kerberos Tokens. This <i>ConformanceUnit</i> only applies if the "Security User IssuedToken Kerberos" is supported.	
Security	Security User Anonymous	The Server provides support for Anonymous access. The use of this feature must be able to be enabled or disabled by an Administrator. By default Anonymous access shall be disabled.	
Security	Security User IssuedToken Kerberos Client	A <i>Client</i> uses a Kerberos Server token. Specific encryption of the issuedToken is required if no message encryption is used. The use of this token is defined by the Kerberos documentation.	
Security	Security User IssuedToken Kerberos Windows Client	A <i>Client</i> uses the Windows implementation of Kerberos tokens. This <i>ConformanceUnit</i> only applies if the "Security User IssuedToken Kerberos Client" is supported.	
Security	Security User Name Password Client	A <i>Client</i> uses a User Name/Password combination. Encryption of the password with the algorithm provided in the UserNameIdentityToken is required if no message encryption is used.	
Security	Security User X509 Client	A <i>Client</i> uses a public/private key pair for user identity. This includes all validation and trust issues associated with a certificate.	

Table 12 describes protocol and encoding related features that can be profiled. These features are defined in detail in IEC 62541-6. It is recommended that *Servers* and *Clients* support as many of these options as possible for greatest interoperability.

Table 12 – Protocol and Encoding

Category	Title	Description	Derived
Server	Protocol Configuration	Allow administration of the Endpoints and the port number used by the Endpoints.	
Transport	Protocol TCP Binary UA Security	Support the UA TCP transport protocol with UA Binary Encoding and with UA Secure Conversation.	
Transport	Protocol HTTPS with UA Binary	Support the HTTPS protocol with UA Binary Encoding.	
Transport	Protocol HTTPS with Soap	Support the HTTPS protocol with Soap-based Xml Encoding.	
Transport	Protocol Soap Xml WS Security	Support "SOAP/HTTP" transport with XML Encoding and with WS Secure Conversation.	
Transport	Protocol Soap Binary WS Security	Support "SOAP/HTTP" transport with UA Binary Encoding and with WS Secure Conversation.	

5.4 Information Model and AddressSpace related features

Table 13 describes Base features related items that can be profiled. For additional information about these items, please refer to IEC 62541-3, IEC 62541-5 and IEC 62541-6. *Servers* with a larger resource capacity would support most of this functionality, but smaller resource constraint *Server* may omit some of this functionality. Many *Clients* would utilize some of this functionality and more robust *Clients* would utilize most of this functionality.

Withhold

Table 13 – Base information

Category	Title	Description	Derived
Server	Base Info Core Structure	The <i>Server</i> supports the <i>Server Object</i> , <i>ServerCapabilities</i> and supports the OPC UA <i>AddressSpace</i> structure.	
Server	Base Info Server Capabilities	The <i>Server</i> supports publishing of the <i>Server</i> limitation in the <i>ServerCapabilities</i> , including <i>MaxArrayLength</i> , <i>MaxStringLength</i> , <i>MaxNodePerRead</i> , <i>MaxNodesPerWrite</i> , <i>MaxNodesPerSubscription</i> and <i>MaxNodesPerBrowse</i> .	
Server	Base Info Progress Events	The <i>Server</i> exposes if generation of Progress events for long running service calls such as <i>HistoryRead</i> or <i>Query</i> is supported. If it is listed as supported in <i>ServerCapabilities</i> , then the actual events are verified.	
Server	Base Info Diagnostics	The <i>Server</i> supports Diagnostic <i>Objects</i> and <i>Variables</i> .	
Server	Base Info System Status	The <i>Server</i> supports generating <i>SystemStatusChangeEvent</i> indicating shutdown of the <i>Server</i> (<i>SourceNode=Server</i>).	
Server	Base Info System Status underlying system	The <i>Server</i> supports generating <i>SystemStatusChangeEvent</i> indicating changes to an underlying system (<i>SourceNode=Server</i>). This event can also be used to indicate that the OPC UA <i>Server</i> has underlying systems.	
Server	Base Info GetMonitoredItems Method	The <i>Server</i> supports obtaining subscription information via <i>GetMonitoredItems Method</i> on the <i>Server</i> object.	
Server	Base Info Type System	The <i>Server</i> exposes a Type System with <i>DataTypes</i> , <i>ReferenceTypes</i> , <i>ObjectTypes</i> and <i>VariableTypes</i> including all of the OPC UA (namespace 0) types that are used by the <i>Server</i> , as defined in IEC 62541-5. Items that are defined in Namespace 0 but are defined in other specification parts are tested as part of the other information models.	

Category	Title	Description	Derived
Server	Base Info Custom Type System	The <i>Server</i> supports defining user defined <i>ObjectTypes</i> , <i>VariableTypes</i> , <i>ReferenceType</i> and <i>DataTypes</i> . Supporting this conformance unit does not require that a <i>Server</i> exposes the OPC UA <i>Object</i> , <i>Variable</i> , <i>Reference</i> , or <i>DataTypes</i> , unless the <i>Server</i> implements User types. If User types are defined then the full type-hierarchy has to be exposed as well.	
Server	Base Info Model Change	The <i>Server</i> supports <i>ModelChange Event</i> and <i>NodeVersion Property</i> for all <i>Nodes</i> that the server allows <i>Model changes</i> for.	
Server	Base Info Placeholder Modelling Rules	The <i>Server</i> supports defining custom <i>Object</i> or <i>Variables</i> that include the use of <i>OptionalPlaceholder</i> or <i>MandatoryPlaceholder</i> modelling rules.	
Server	Base Info SemanticChange	The <i>Server</i> supports <i>SemanticChangeEvent</i> for some <i>Properties</i> . This includes setting the <i>SemanticChange</i> Bit in the status when a semantic change occurs, such as a change in the engineering unit associated with a value.	
Server	Base Info EventQueueOverflowEventType	The <i>Server</i> supports the <i>EventQueueOverflowEventType</i> as defined in IEC 62541-4.	
Server	Base Info OptionSet	The <i>Server</i> supports the <i>VariableType OptionSet</i> .	
Server	Base Info ValueAsText	The <i>Server</i> supports the <i>Property ValueAsText</i> for enumerated <i>DataTypes</i> .	
Server	Base Info Engineering Units	The <i>Server</i> supports defining <i>Variables</i> that include the <i>Engineering Units Property</i> . This property makes use of the <i>EUInformation</i> data structure. This structure by default represents the UN/CEFACT "Codes for Units of Measurement". If a different EU representation is required then the <i>EUInformation.namespaceUri</i> will indicate the alternate namespace.	
Server	Base Info FileType Base	The <i>Server</i> supports the <i>FileType Object</i> (see IEC 62541-5). File writing may be restricted.	
Server	Base Info FileType Write	The <i>Server</i> supports the <i>FileType Object</i> , including writing of files. Also included is the support of user access control on <i>FileType Object</i> .	

Category	Title	Description	Derived
Client	Base Info Client Basic	The <i>Client</i> uses the defined OPC UA <i>AddressSpace</i> . Access or provide access to <i>Server</i> information like the <i>Server's</i> state, <i>BuildInfo</i> , capabilities, <i>Namespace Table</i> and <i>Type Model</i> .	
Client	Base Info Client System Status	The <i>Client</i> makes use of <i>SystemStatusChangeEvent</i> to detect server shutdowns.	
Client	Base Info Client Progress Events	The <i>Client</i> makes use of <i>ProgressEvents</i> , including checking for their support.	
Client	Base Info Client Diagnostics	The <i>Client</i> provides interactive or programmatic access to the <i>Server's</i> diagnostic information.	
Client	Base Info Client Type Programming	The <i>Client</i> programmatically process instances of <i>Objects</i> or <i>Variables</i> by using their type definitions. This includes custom <i>DataTypes</i> , <i>ObjectTypes</i> and <i>VariableTypes</i> .	
Client	Base Info Client Change Events	The <i>Client</i> processes <i>ModelChangeEvents</i> to detect changes in the <i>Server's</i> OPC UA <i>AddressSpace</i> and take appropriate action for a given change.	
Client	Base Info Client GetMonitoredItems Method	The <i>Client</i> makes use of <i>GetMonitoredItems Method</i> to recover for communication interruptions and/or to recover subscription information.	
Client	Base Info Client FileType Base	The <i>Client</i> can access a <i>FileType Object</i> to transfer a file from the <i>Server</i> to the <i>Client</i> . This includes large files.	
Client	Base Info Client FileType Write	The <i>Client</i> can access a <i>FileType Object</i> to transfer a file from the <i>Client</i> to the <i>Server</i> . This includes large files.	

Table 14 describes Address Space Model information related items that can be profiled. The details of these model items are defined in IEC 62541-3 and IEC 62541-5. This include *Server Facets* that describe what a *Server* exposes and *Client Facets* that describe what a *Client* consumes

Table 14 – Address Space model

Category	Title	Description	Derived
Server	Address Space Base	Support the <i>NodeClasses</i> with their <i>Attributes</i> and behaviour as defined in IEC 62541-3. This includes for instance: <i>Object</i> , <i>ObjectType</i> , <i>Variable</i> , <i>VariableType</i> , <i>References</i> and <i>DataType</i> .	
Server	Address Space Events	Support OPC UA <i>AddressSpace</i> elements for generating <i>Event</i> notifications. This includes at least one <i>Node</i> with an <i>EventNotifier Attribute</i> set to True (<i>Server Node</i>).	
Server	Address Space Complex DataTypes	Support <i>StructuredDataTypes</i> with a <i>Data Dictionary</i> .	
Server	Address Space Method	Support <i>Method Nodes</i> .	
Server	Address Space Notifier Hierarchy	Supports using the <i>HasNotifier</i> reference to build a hierarchy of <i>Object Nodes</i> that are notifiers with other notifier <i>Object Nodes</i> .	
Server	Address Space Source Hierarchy	Supports hierarchies of event sources where each hierarchy roots in an <i>Object Node</i> that is a notifier. The <i>HasEventSource Reference</i> is used to relate the <i>Nodes</i> within a hierarchy. If <i>Conditions</i> are supported, the hierarchy shall include <i>HasCondition References</i> .	
Server	Address Space WriteMask	Supports <i>WriteMask</i> indicating the write access availability for all attributes, including not supported attributes.	
Server	Address Space UserWriteMask	Supports <i>UserWriteMask</i> indicating the write access availability for all attributes for the given user, including not supported attributes. Support includes at least two levels of users.	
Server	Address Space UserWriteMask Multilevel	Supports <i>UserWriteMask</i> indicating the write access availability for all attributes for the given user, including not supported attributes. This includes supporting multiple levels of access control for all nodes in the system.	
Server	Address Space User Access Level Full	Implements User Access Level security, this includes supporting multiple levels of access control for <i>Variable</i> nodes in the system. This includes an indication of read, write, Historical read and Historical write access to the <i>Value Attribute</i> .	
Server	Address Space User Access Level Base	Implements User Access Level Security for <i>Variable</i> nodes, this includes at least two users in the system. This includes an indication of read, write, historical read and Historical write access to the value attribute	
Client	Address Space Client Base	Uses and understands the <i>NodeClasses</i> with their <i>Attributes</i> and behaviour as defined in IEC 62541-3. This includes for instance: <i>Object</i> , <i>ObjectType</i> , <i>Variable</i> , <i>VariableType</i> , <i>References</i> and <i>DataType</i> . This includes treating <i>BrowseNames</i> and <i>String NodeIds</i> as case sensitive.	

Category	Title	Description	Derived
Client	Address Space Client Complex DataTypes	Uses and understands arbitrary StructuredDataTypes via Data Dictionary.	
Client	Address Space Client Notifier Hierarchy	Uses hierarchy of <i>Object Nodes</i> that are notifiers to detect specific areas where the <i>Client</i> can subscribe for Events.	
Client	Address Space Client Source Hierarchy	Detect and use the hierarchy of event sources exposed for specific <i>Object Nodes</i> that are event notifiers.	

Table 15 describes Data Access information model related items that can be profiled. The details of this model are defined in IEC 62541-8. *Server* could expose this information model and *Client* could utilize this information model.

Withdrawn

Table 15 – Data Access

Category	Title	Description	Derived
Server	Data Access DataItems	Provide <i>Variables</i> of <i>DataItem</i> Type or one of its subtypes. Support the <i>StatusCodes</i> specified in the IEC 62541-8. Support of optional <i>Properties</i> (e.g. "InstrumentRange") shall be verified during certification testing and will be shown in the <i>Certificate</i> .	
Server	Data Access AnalogItems	Support <i>AnalogItem</i> Type <i>Variables</i> with corresponding <i>Properties</i> . The support of optional <i>properties</i> will be listed.	
Server	Data Access PercentDeadband	Support <i>PercentDeadband</i> filter when monitoring <i>AnalogItem</i> Type <i>Variables</i> .	
Server	Data Access Semantic Changes	Support semantic changes of <i>AnalogItem</i> Type items (<i>EURange</i> <i>Property</i> and/or <i>EngineeringUnits</i> <i>Property</i>). Support semantic change <i>StatusCode</i> bits where appropriate.	
Server	Data Access TwoState	Support <i>TwoStateDiscrete</i> Type <i>Variables</i> with corresponding <i>Properties</i> .	
Server	Data Access MultiState	Support <i>MultiStateDiscrete</i> Type <i>Variables</i> with corresponding <i>Properties</i> .	
Server	Data Access ArrayItemType	Provide <i>Variables</i> of <i>ArrayItem</i> Type or one of its subtypes (<i>YArrayItem</i> Type, <i>XYArrayItem</i> Type, <i>ImageArray</i> Type, <i>CubeArray</i> Type and <i>NDimensionArray</i> Type). The supported subtypes will be listed. Support for this type includes supporting all of the mandatory <i>properties</i> including <i>AxisInformation</i> .	
Server	Data Access Complex Number	Supports the <i>Complex Number</i> data type. This data type is available for any variable types that do not have other explicit restrictions.	
Server	Data Access DoubleComplex Number	Supports the <i>DoubleComplex Number</i> data type. This data type is available for any variable types that do not have other explicit restrictions.	
Client	Data Access Client Basic	Understand the <i>DataAccess</i> <i>Variable</i> Types. Make use of the standard <i>Properties</i> if applicable.	
Client	Data Access Client Deadband	Use <i>PercentDeadband</i> to filter value changes of <i>AnalogItem</i> Type <i>Variables</i> .	
Client	Data Access Client SemanticChange	Recognize the semantic change bit in the <i>StatusCode</i> while monitoring items and take proper action. Typically, the <i>Client</i> has to re-read <i>Properties</i> that define type-specific semantic like the <i>EURange</i> and <i>EngineeringUnits</i> <i>Properties</i> .	

Table 16 describes *Alarm* and *Conditions* information model related items that can be profiled. The details of this model are defined in IEC 62541-9. *Servers* that deal with *Alarm* and *Conditions* would expose this information model and *Clients* that process *Alarms* and *Conditions* would utilize this information model.

Table 16 – Alarms and Conditions

Category	Title	Description	Derived
Server	A & C Basic	Supports <i>Alarm</i> & <i>Condition</i> model <i>ConditionType</i> .	
Server	A & C Enable	Supports Enable and Disable Methods.	
Server	A & C Refresh	Supports <i>ConditionRefresh Method</i> and the concept of a refresh.	
Server	A & C Instances	Support the exposing of A&C <i>Conditions</i> in the <i>AddressSpace</i> .	
Server	A & C ConditionClasses	Supports multiple <i>Condition</i> classes for grouping and filtering of <i>Alarms</i> .	
Server	A & C Acknowledge	Support Acknowledge, includes Acknowledge <i>Method</i> , Acknowledgeable type.	
Server	A & C Confirm	Support confirming <i>Conditions</i> , includes Confirm method.	
Server	A & C Comment	Support Comments, includes AddComment <i>Method</i> .	
Server	A & C Alarm	Support for Basic <i>Alarm</i> functionality, including active, inactive states.	
Server	A & C Branch	Support for <i>Alarm</i> Branches which includes previous <i>Condition</i> Instances, i.e. conditions instance other than the current condition that still requires some operator action, such as acknowledgement or a dialog.	
Server	A & C Shelving	Support for the shelving mode, including the TimedShelve, OneShotShelve and Unshelve methods.	
Server	A & C Exclusive Level	Supports Exclusive Level <i>Alarm</i> type.	
Server	A & C Exclusive Limit	Supports Exclusive Limit <i>Alarms</i> . A <i>Server</i> that supports this must support one of the sub-types Level, Deviation or RateofChange.	
Server	A & C Exclusive Deviation	Supports Exclusive Deviation <i>Alarm</i> type.	
Server	A & C Exclusive RateofChange	Supports Exclusive RateofChange <i>Alarm</i> type.	
Server	A & C Non-Exclusive Limit	Supports Non-Exclusive Limit <i>Alarms</i> . A <i>Server</i> that supports this must support one of the sub-types Level, Deviation or RateofChange.	
Server	A & C Non-Exclusive Level	Supports Non-Exclusive Level <i>Alarm</i> type.	
Server	A & C Non-Exclusive Deviation	Supports Non-Exclusive Deviation <i>Alarm</i> type.	
Server	A & C Non-Exclusive RateofChange	Supports Non-Exclusive RateofChange <i>Alarm</i> type.	
Server	A & C Discrete	Supports Discrete <i>Alarm</i> types.	
Server	A & C Off Normal	Supports Off Normal <i>Alarm</i> type.	
Server	A & C Trip	Supports Trip <i>Alarm</i> type.	
Server	A & C Dialog	Supports DialogConditionType including Respond <i>Method</i> .	
Server	A & E Wrapper Mapping	The <i>Server</i> uses the COM A&E mapping specified in IEC 62541-9 to map COM Events to A&C Events, this includes <i>Condition</i> Class mapping.	

Category	Title	Description	Derived
Client	A & C Basic Client	Uses the <i>Alarm & Condition</i> model <i>ConditionType</i> .	
Client	A & C Enable Client	Uses Enable and Disable Methods.	
Client	A & C Refresh Client	Uses <i>ConditionRefresh Method</i> and the concept of a refresh.	
Client	A & C Instances Client	Uses <i>A&C Conditions</i> that are exposed in the <i>AddressSpace</i> .	
Client	A & C ConditionClasses Client	Uses <i>Condition</i> classes to group <i>Alarms</i> .	
Client	A & C Acknowledge Client	Uses Acknowledge, including <i>Acknowledge Method</i> , <i>Acknowledgeable</i> type.	
Client	A & C Confirm Client	Uses confirming <i>Conditions</i> , including <i>Confirm method</i> .	
Client	A & C Comment Client	Uses Comments, including <i>AddComment Method</i> .	
Client	A & C Alarm Client	Uses Basic <i>Alarm</i> functionality, including active, inactive states.	
Client	A & C Branch Client	Uses <i>Alarm Branches</i> which included previous <i>Condition</i> Instances, i.e. conditions instance other than the current condition that still requires some action, such as acknowledgement or confirmation.	
Client	A & C Shelving Client	Uses the shelving model, including the <i>TimedShelve</i> , <i>OneShotShelve</i> and <i>Unshelve</i> methods.	
Client	A & C Exclusive Level Client	Uses Exclusive Level <i>Alarms</i> as defined.	
Client	A & C Exclusive Limit Client	Uses Exclusive Limit <i>Alarms</i> . Requires that at least one of the sub-types be used.	
Client	A & C Exclusive Deviation Client	Uses Exclusive Deviation <i>Alarms</i> .	
Client	A & C Exclusive RateofChange Client	Uses Exclusive RateofChange <i>Alarms</i> .	
Client	A & C Non-Exclusive Level Client	Uses Non-Exclusive Level <i>Alarms</i> .	
Client	A & C Non-Exclusive Limit Client	Uses Non-Exclusive Limit <i>Alarms</i> . Requires that at least one of the sub-types be used.	
Client	A & C Non-Exclusive Deviation Client	Uses Non-Exclusive Deviation <i>Alarms</i> .	
Client	A & C Non-Exclusive RateofChange Client	Uses Non-Exclusive RateofChange <i>Alarms</i> .	
Client	A & C Discrete Client	Uses Discrete <i>Alarm</i> types.	
Client	A & C Off Normal Client	Uses the Off Normal <i>Alarm</i> types.	
Client	A & C Trip Client	Uses the Trip <i>Alarm</i> type.	
Client	A & C Dialog Client	Uses the <i>DialogConditionType</i> including <i>Respond Method</i> .	

Table 17 describes Historical Data Access information model related items that can be profiled. The details of this model are defined in IEC 62541-11. *Servers* that support some level of historical data would expose this information model and *Clients* that utilize historical data would utilize this information model.

Table 17 – Historical Access

Category	Title	Description	Derived
Server	Historical Access Read Raw	General support for basic historical access, reading raw data using the ReadRawModifiedDetails structure. Where the time range is specified using a start time, stop time and number of values (a minimum of two of the three parameters must be provided) and the ReadModified flag is set to False.	
Server	Historical Access Data Max Nodes Read Continuation Point	Supports enough continuation points to cover the number of supported points indicated in the MaxNodesPerHistoryReadData Server OperationLimits parameter for historical data access.	
Server	Historical Access Time Instance	Supports reading historical data at a specified instance in time using the ReadAtTimeDetails structure.	
Server	Historical Access Aggregates	Supports reading one or more Aggregates of historical values of <i>Variables</i> using the ReadProcessedDetails structure. At least one of the Aggregates described in IEC 62541-13 must be supported. The complete list will be shown in the <i>Software Certificate</i> .	
Server	Historical Access Insert Value	Supports inserting historical values of <i>Variables</i> .	
Server	Historical Access Delete Value	Supports deleting historical values of <i>Variables</i> .	
Server	Historical Access Update Value	Supports updating historical values of <i>Variables</i> .	
Server	Historical Access Replace Value	Supports replacing historical values of <i>Variables</i> .	
Server	Historical Access Modified Values	Supports maintaining old values for historical data that have been updated and the retrieval of these values using the ReadRawModifiedDetails structure (ReadModified flag set to true).	
Server	Historical Access Annotations	Supports the entry and retrieval of Annotations for historical data. The retrieval is accomplished using the standard historical read raw functionality (ReadRawModifiedDetails). The entry uses the standard historical update (UpdateStructureDataDetails) functionality.	
Server	Historical Access ServerTimestamp	Supports providing a ServerTimestamp (as well as the default SourceTimestamp).	
Server	Historical Access Structured Data Read Raw	Supports ReadRawModified historical access for structured data. Supporting the structure for an annotation is not considered supporting generic structured data.	

Category	Title	Description	Derived
Server	Historical Access Structured Data Time Instance	Supports historical access for structured data. Supporting ReadAtTimeDetails for structured data. Supporting the structure for an annotation is not considered supporting generic structured data.	
Server	Historical Access Structured Data Insert	Supports historical access for structured data. Inserting Structured data. Supporting the structure for an annotation is not considered supporting generic structured data.	
Server	Historical Access Structured Data Delete	Supports historical access for structured data. Delete of existing data. Supporting the structure for an annotation is not considered supporting generic structured data.	
Server	Historical Access Structured Data Update	Supports historical access for structured data. Updates of existing data. Supporting the structure for an annotation is not considered supporting generic structured data.	
Server	Historical Access Structured Data Replace	Supports replacing structured historical data. Supporting the structure for an annotation is not considered supporting generic structured data.	
Server	Historical Access Structured Data Read Modified	Supports maintaining old values for historical structured data that have been updated and the retrieval of these values. Using the ReadRawModifiedDetails structure (ReadModified flag set to true) for structured data. Supporting the structure for an annotation is not considered supporting generic structured data.	
Server	Historical Access Events	Supports the retrieval of historical Events using the ReadEventDetails structure. This includes support for simple filtering of Events. The <i>Event</i> fields that are stored are server specific, but at least the mandatory fields of BaseEventType are required.	
Server	Historical Access Event Max Events Read Continuation Point	Supports enough continuation points to cover the number of supported <i>Event</i> reads indicated in the MaxNodesPerHistoryReadEvents Server OperationLimits parameter for Historical <i>Event</i> access.	
Server	Historical Access Insert Event	Supports inserting historical Events.	
Server	Historical Access Update Event	Supports updating historical Events.	
Server	Historical Access Replace Event	Supports replacing historical Events.	
Server	Historical Access Delete Event	Supports deleting of historical Events.	
Client	Historical Access Client Browse	Uses the View <i>Service</i> Set to discover <i>Nodes</i> with historical data.	
Client	Historical Access Client Read Raw	Uses the HistoryRead <i>Service</i> to read raw historical data using the ReadRawModifiedDetails Structure (ReadModified Flag set to False).	

Category	Title	Description	Derived
Client	Historical Access Client Read Modified	Uses the HistoryRead <i>Service</i> to read modified historical data using the ReadRawModifiedDetails Structure (ReadModified Flag set to True).	
Client	Historical Access Client Read Aggregates	Uses the HistoryRead <i>Service</i> to read Aggregated historical data. This includes using at least one of the Aggregates defined in IEC 62541-13. The complete list of Aggregates used by the <i>Client</i> is included in the results of this <i>ConformanceUnit</i> .	
Client	Historical Access Client Structure Data Raw	Uses the HistoryRead <i>Service</i> to read raw historical data using the ReadRawModifiedDetails Structure (ReadModified Flag set to False) for structured data.	
Client	Historical Access Client Structure Data Read Modified	Uses the HistoryRead <i>Service</i> to read modified structured historical data using the ReadRawModifiedDetails Structure (ReadModified Flag set to True).	
Client	Historical Access Client Structure Data Insert	Uses the HistoryUpdate <i>Service</i> to insert historical data values for structured data.	
Client	Historical Access Client Structure Data Delete	Uses the HistoryUpdate <i>Service</i> to delete historical data values for structured data.	
Client	Historical Access Client Structure Data Update	Uses the HistoryUpdate <i>Service</i> to update historical data values for structured data.	
Client	Historical Access Client Structure Data Replace	Uses the HistoryUpdate <i>Service</i> to replace historical data values for structured data.	
Client	Historical Access Client Structure Data Time Instance	Reads historical data at a specified instance in time for structured data. Using the ReadAtTimeDetails structure.	
Client	Historical Access Client Read Events	Uses the HistoryRead <i>Service</i> to read historical <i>Event</i> data using the ReadEventDetails Structure.	
Client	Historical Access Client Event Inserts	Uses the HistoryUpdate <i>Service</i> to insert historical Events.	
Client	Historical Access Client Event Updates	Uses the HistoryUpdate <i>Service</i> to update historical Events.	
Client	Historical Access Client Event Replaces	Uses the HistoryUpdate <i>Service</i> to replace historical Events.	
Client	Historical Access Client Event Deletes	Uses the HistoryUpdate <i>Service</i> to delete historical Events.	
Client	Historical Access Client Data Insert	Uses the HistoryUpdate <i>Service</i> to insert historical data values.	
Client	Historical Access Client Data Delete	Uses the HistoryUpdate <i>Service</i> to delete historical data values.	
Client	Historical Access Client Data Update	Uses the HistoryUpdate <i>Service</i> to update historical data values.	

Category	Title	Description	Derived
Client	Historical Access Client Data Replace	Uses the HistoryUpdate <i>Service</i> to replace historical data values.	
Client	Historical Access Client Annotations	Enters and retrieves Annotations of historical data. The retrieval is accomplished using the standard historical read raw functionality (ReadRawModifiedDetails). The entry uses the standard Historical Update (UpdateStructureDataDetails) functionality.	
Client	Historical Access Client Time Instance	Reads historical data at a specified instance in time using the ReadAtTimeDetails structure.	
Client	Historical Access Client Server Timestamp	Uses the ServerTimestamp (as well as the default SourceTimestamp), if it is provided by the <i>Server</i> .	

Table 18 describes Aggregate related items that can be profiled. Servers that support the Aggregates would expose this functionality and *Clients* that utilize Aggregates would implement some of this functionality.

Withdawn