

DIN EN IEC 62541-7

ICS 35.200; 35.240.50

Ersatz für
DIN EN 62541-7:2015-11
Siehe Anwendungsbeginn

**OPC Unified Architecture –
Teil 7: Profile
(IEC 62541-7:2020);
Englische Fassung EN IEC 62541-7:2020**

OPC unified architecture –
Part 7: Profiles
(IEC 62541-7:2020);
English version EN IEC 62541-7:2020

Architecture unifiée OPC –
Partie 7: Profils
(IEC 62541-7:2020);
Version anglaise EN IEC 62541-7:2020

Gesamtumfang 136 Seiten

DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE

Anwendungsbeginn

Anwendungsbeginn für die von CENELEC am 2020-07-27 angenommene Europäische Norm als DIN-Norm ist 2021-08-01.

Für DIN EN 62541-7:2015-11 besteht eine Übergangsfrist bis 2023-07-27.

Nationales Vorwort

Vorausgegangener Norm-Entwurf: E DIN EN IEC 62541-7:2019-09.

Für dieses Dokument ist das nationale Arbeitsgremium DKE/K 931 „Systemaspekte der Automatisierung“ der DKE Deutsche Kommission Elektrotechnik Elektronik Informationstechnik in DIN und VDE (www.dke.de) zuständig.

Aktuelle Informationen zu diesem Dokument können über die Internetseiten von DKE (www.dke.de) und DIN (www.din.de) durch eine Suche nach der Dokumentennummer aufgerufen werden.

Die enthaltene IEC-Publikation wurde vom IEC/SC 65E „Devices and integration in enterprise systems“ erarbeitet und unterliegt dem Copyright der IEC .

Das IEC-Komitee hat entschieden, dass der Inhalt dieses Dokuments bis zu dem Datum (stability date) unverändert bleiben soll, das auf der IEC-Website unter „<http://webstore.iec.ch>“ zu diesem Dokument angegeben ist. Zu diesem Zeitpunkt wird entsprechend der Entscheidung des Komitees das Dokument

- bestätigt,
- zurückgezogen,
- durch eine Folgeausgabe ersetzt oder
- geändert.

In diesem Dokument und in den anderen Teilen der IEC 62541-Reihe werden bestimmte Dokumentenkonventionen verwendet:

Kursivschrift wird verwendet, um einen definierten Begriff oder eine Definition zu kennzeichnen, der/die in dem Abschnitt „Begriffe und Abkürzungen“ in einem der Teile der IEC 62541-Reihe erscheint.

Kursivschrift wird auch verwendet, um den Namen eines Service-Eingangs- oder -Ausgangsparameters oder den Namen einer Struktur oder eines Elements einer Struktur zu kennzeichnen, die normalerweise in Tabellen definiert sind.

Die *kursiv* gedruckten Begriffe und Namen werden mit wenigen Ausnahmen in Camel-Case-Schreibweise geschrieben (die Schreibweise von zusammengesetzten Wörtern oder Phrasen, in denen die Elemente ohne Leerzeichen verbunden sind, wobei der Anfangsbuchstabe jedes Elements innerhalb der Verbindung groß geschrieben wird). Zum Beispiel lautet der definierte Begriff „AddressSpace“ statt „Address Space“. Dies macht es einfacher zu verstehen, dass es eine einzige Definition für AddressSpace gibt und nicht separate Definitionen für Address und Space.

Für den Fall einer undatierten Verweisung im normativen Text (Verweisung auf ein Dokument ohne Angabe des Ausgabedatums und ohne Hinweis auf eine Abschnittsnummer, eine Tabelle, ein Bild usw.) bezieht sich die Verweisung auf die jeweils aktuellste Ausgabe des in Bezug genommenen Dokuments.

Für den Fall einer datierten Verweisung im normativen Text bezieht sich die Verweisung immer auf die in Bezug genommene Ausgabe des Dokuments.

Der Zusammenhang der zitierten Dokumente mit den entsprechenden deutschen Dokumenten ergibt sich, soweit ein Zusammenhang besteht, grundsätzlich über die Nummer der entsprechenden IEC-Publikation.

Beispiel: IEC 60068 ist als EN 60068 als Europäische Norm durch CENELEC übernommen und als DIN EN 60068 ins Deutsche Normenwerk aufgenommen.

Das Präsidium des DIN hat mit Präsidialbeschluss 1/2004 festgelegt, dass DIN-Normen, deren Inhalt sich auf internationale Arbeitsergebnisse der Informationsverarbeitung gründet, unter bestimmten Bedingungen allein in englischer Sprache veröffentlicht werden dürfen. Diese Bedingungen sind für die vorliegende Norm erfüllt.

Da sich die Benutzer der vorliegenden Norm der englischen Sprache als Fachsprache bedienen, wird die Englische Fassung der EN IEC 62541-7 veröffentlicht. Zu deren Abschnitt 1, der den Anwendungsbereich festlegt und Abschnitt 3 (außer Unterabschnitt 3.2), der die Begriffe festlegt, wurde eine Übersetzung angefertigt und als informativer Nationaler Anhang NA der vorliegenden Norm hinzugefügt. Für die meisten der verwendeten Begriffe existieren keine gebräuchlichen deutschen Benennungen, da sich die deutschen Anwender in der Regel ebenfalls der englischen Benennungen bedienen. Diese Norm steht nicht in unmittelbarem Zusammenhang mit Rechtsvorschriften und ist nicht als Sicherheitsnorm anzusehen.

Das Original-Dokument enthält Bilder in Farbe, die in der Papierversion in einer Graustufen-Darstellung wiedergegeben werden. Elektronische Versionen dieses Dokuments enthalten die Bilder in der originalen Farbdarstellung.

Änderungen

Gegenüber DIN EN 62541-7:2015-11 wurden folgende Änderungen vorgenommen:

- a) Neue Funktionsprofile:
 - Profile für globales Discovery- und globales Zertifikatsmanagement;
 - Profile für das Management globaler KeyCredential und das globale Zugangs-Token-Management;
 - Facette für dauerhafte Subskriptionen;
 - Standard-UA-Client-Profil;
 - Profile für die Verwaltung von Benutzerrollen und -genehmigungen;
- b) Neue Transportprofile:
 - HTTPS mit JSON-Kodierung;
 - sichere WebSockets (WSS) mit binärer JSON-Kodierung;
 - Rückwärtskonnektivität;
- c) Neue Sicherheitsprofile:
 - TransportSicherheit – TLS 1.2 mit PFS (mit Perfect Forward Secrecy);
 - SicherheitsRichtlinie [A] – Aes128-Sha256-RsaOaep (ersetzt Basic128Rsa15);
 - SicherheitsRichtlinie – Aes256-Sha256-RsaPss fügt Perfect Forward Secrecy für UA-TCP hinzu);
 - Benutzer-Token-JWT (Jason-Web-Token);
- d) Veraltete Sicherheitsprofile (aufgrund defekter Algorithmen):
 - SicherheitsRichtlinie – Basic128Rsa15 (defekter Algorithmus Sha1);
 - SicherheitsRichtlinie – Basic256 (defekter Algorithmus Sha1);
 - TransportSicherheit – TLS 1.0 (defekter Algorithmus RC4);
 - TransportSicherheit – TLS 1.1 (defekter Algorithmus RC4);
- e) Veralteter Transport (fehlende Unterstützung bei den meisten Plattformen):
 - SOAP/HTTP mit WS-SecureConversation (alle Kodierungen).

Frühere Ausgaben

DIN EN 62541-7: 2013-06, 2015-11

Nationaler Anhang NA (informativ)

Auszugsweise Übersetzung

Reihenfolge und Inhalt der folgenden Abschnitte sind identisch mit denen in den jeweiligen Abschnitten der Englischen Fassung.

1 Anwendungsbereich

Dieser Teil der Normenreihe IEC 62541 legt die *Profile* der OPC Unified Architecture (OPC-UA) (vereinheitlichte Architektur der plattformunabhängigen Kommunikation) fest. Die *Profile* in diesem Dokument werden verwendet, um die Eigenschaften bezüglich Prüfungen von OPC-UA-Produkten und den Charakter der Prüfungen (basierend auf Werkzeugen oder basierend auf einem Prüflabor) zu trennen. Dies umfasst die Prüfungen, welche durch das von der OPC-Organisation bereitgestellte OPC-UA-CTT (ein Selbstprüfungswerkzeug) und durch von der OPC-Organisation bereitgestellte unabhängige Prüflaboratorien zur Zertifizierung durchgeführt werden. Dies könnte sich gleichermaßen ebenso auf Prüfwerkzeuge beziehen, welche von einer anderen Organisation oder einem Prüflaboratorium einer anderen Organisation bereitgestellt werden. Wichtig ist das Konzept der automatisierten Prüfung basierend auf Werkzeugen im Gegensatz zur Prüfung basierend auf einem Laboratorium. Der Anwendungsbereich dieser Norm umfasst die Festlegung der Funktionalitäten, welche nur in einem Prüflaboratorium geprüft werden kann, und der Festlegung der Gruppierung von Funktionalität, welche bei der Prüfung von OPC-UA-Produkten entweder in einem Laboratorium oder mit automatisierten Werkzeugen zu prüfen ist. Die Festlegung von tatsächlichen *PrüfFällen* (en: TestCases) liegt nicht im Anwendungsbereich dieses Dokuments, aber die allgemeinen Kategorien von PrüfFällen liegen im Anwendungsbereich dieses Dokuments.

Die meisten OPC-UA-Anwendungen stimmen mit einigen, aber nicht allen, dieser *Profile* überein.

3 Begriffe und Abkürzungen

3.1 Begriffe

Für die Anwendung dieses Dokuments gelten die Begriffe nach IEC TR 62541-1, IEC TR 62541-2, IEC 62541-3, IEC 62541-4, IEC 62541-6, IEC 62541-8 und die folgenden Begriffe.

ISO und IEC stellen terminologische Datenbanken für die Anwendung in der Normung unter den folgenden Adressen bereit:

- IEC Electropedia: verfügbar unter <http://www.electropedia.org/>
- ISO Online Browsing Platform: verfügbar unter <http://www.iso.org/obp>

ANMERKUNG: Bild 1 beinhaltet eine Übersicht der in diesem Dokument definierten Begriffe und ihrer Korrelationen.

3.1.1

Anwendung

(en: Application)

Programm, das Aspekte von OPC-UA umsetzt oder bearbeitet

Anmerkung 1 zum Begriff: Die Anwendung kann auf jeder Maschine laufen und jegliche Funktion ausführen. Die Anwendung kann eine Software- oder eine Hardware-Anwendung sein, die einzige Anforderung ist, dass sie OPC-UA umsetzt.

3.1.2

KonformitätsEinheit

(en: ConformanceUnit)

definierte Menge von OPC-UA-Eigenschaften, die als Einheit geprüft werden können

Anmerkung 1 zum Begriff: Eine KonformitätsEinheit kann eine Gruppe von Diensten, Teile von Diensten oder Informationsmodelle abdecken.

3.1.3

KonformitätsGruppe

(en: ConformanceGroup)

Gruppierung von KonformitätsEinheiten mit eigenständigen Namen

Anmerkung 1 zum Begriff: Die Gruppierung dient lediglich der Organisation von KonformitätsEinheiten (en: ConformanceUnits). Typische KonformitätsGruppen beinhalten Gruppen für jeden Satz von Diensten in OPC-UA und jede der Informationsmodell-Spezifikationen.

3.1.4

Facette

(en: Facet)

Profil, welches einer spezifischen Eigenschaft zugeordnet ist, die ein Server oder Client benötigen kann

Anmerkung 1 zum Begriff: Facetten werden typischerweise zusammengefasst, um ein Profil einer höheren Ebene zu bilden. Die Verwendung des Begriffs Facette im Titel eines Profils zeigt an, dass das Profil nicht alleine eingesetzt werden kann.

3.1.5

VollAusgestattetes Profil

(en: FullFeatured Profile)

Profil, das alles festlegt, was zum Aufbau einer funktionsfähigen OPC-UA-Anwendung notwendig ist

Anmerkung 1 zum Begriff: Ein VollAusgestattetes Profil fügt insbesondere Definitionen zu Transport- und Informationssicherheitsanforderungen hinzu.

3.1.6

ProfilKategorie

(en: ProfileCategory)

Kategorie, die Profile in Anwendungsklassen, wie z. B. Server oder Client arrangiert

Anmerkung 1 zum Begriff: Diese Kategorien helfen, den Typ der Anwendung herauszufinden, in dem ein Profil verwendet werden kann. Weitere Details werden in 4.4 beschrieben.

3.1.7

PrüfFall

(en: TestCase)

technische Beschreibung einer Menge von Schritten, die erforderlich sind, um eine bestimmte Funktion oder ein bestimmtes Informationsmodell zu testen

Anmerkung 1 zum Begriff: Prüffälle bieten ausreichende Einzelheiten, um einem Entwickler die Umsetzung in Code zu ermöglichen. Prüffälle bieten auch eine genaue Zusammenfassung des (der) erwarteten Ergebnisse(s) bei der Ausführung der Implementierung und alle Vorbedingungen, die erfüllt werden müssen, bevor der PrüfFall ausgeführt werden kann.

3.1.8

PrüfLabor

(en: TestLab)

Ort, der dafür ausgestattet ist, Prüfdienste bereitzustellen

Anmerkung 1 zum Begriff: Diese Dienste umfassen direkt prüfendes Personal, automatisierte Prüfungen und einen formal wiederholbaren Prozess, sind aber nicht darauf beschränkt. Die OPC-Organisation stellt detaillierte Spezifikationen bereit, welche OPC-UA-Prüflaboratorien festlegen sowie die Prüfungen festlegen, welche diese durchführen müssen (siehe auch Compliance Part 8 UA Server, Compliance Part 9 UA Client).

– Leerseite –

English Version

**OPC unified architecture - Part 7: Profiles
(IEC 62541-7:2020)**

Architecture unifiée OPC - Partie 7: Profils
(IEC 62541-7:2020)

OPC Unified Architecture - Teil 7: Profile
(IEC 62541-7:2020)

This European Standard was approved by CENELEC on 2020-07-27. CENELEC members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration.

Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CENELEC member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CENELEC member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CENELEC members are the national electrotechnical committees of Austria, Belgium, Bulgaria, Croatia, Cyprus, the Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, the Netherlands, Norway, Poland, Portugal, Republic of North Macedonia, Romania, Serbia, Slovakia, Slovenia, Spain, Sweden, Switzerland, Turkey and the United Kingdom.



European Committee for Electrotechnical Standardization
Comité Européen de Normalisation Electrotechnique
Europäisches Komitee für Elektrotechnische Normung

CEN-CENELEC Management Centre: Rue de la Science 23, B-1040 Brussels

European foreword

The text of document 65E/707/FDIS, future edition 3 of IEC 62541-7, prepared by SC 65E "Devices and integration in enterprise systems" of IEC/TC 65 "Industrial-process measurement, control and automation" was submitted to the IEC-CENELEC parallel vote and approved by CENELEC as EN IEC 62541-7:2020.

The following dates are fixed:

- latest date by which the document has to be implemented at national level by (dop) 2021-04-27
publication of an identical national standard or by endorsement
- latest date by which the national standards conflicting with the (dow) 2023-07-27
document have to be withdrawn

This document supersedes EN 62541-7:2015 and all of its amendments and corrigenda (if any).

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

This document has been prepared under a mandate given to CENELEC by the European Commission and the European Free Trade Association.

Endorsement notice

The text of the International Standard IEC 62541-7:2020 was approved by CENELEC as a European Standard without any modification.

Annex ZA (normative)

Normative references to international publications with their corresponding European publications

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

NOTE 1 Where an International Publication has been modified by common modifications, indicated by (mod), the relevant EN/HD applies.

NOTE 2 Up-to-date information on the latest versions of the European Standards listed in this annex is available here: www.cenelec.eu.

<u>Publication</u>	<u>Year</u>	<u>Title</u>	<u>EN/HD</u>	<u>Year</u>
IEC/TR 62541-1	–	OPC unified architecture – Part 1: Overview and concepts	CLC/TR 62541-1	–
IEC/TR 62541-2	–	OPC unified architecture – Part 2: Security model	CLC/TR 62541-2	–
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 62541-11	–	OPC Unified Architecture – Part 11:–Historical Access		–
IEC 62541-12	–	OPC unified architecture – Part 12:–Discovery and global services		–
IEC 62541-13	–	OPC Unified Architecture – Part 13:–Aggregates		–
Compliance Part 8 UA Server	–	OPC Test Lab Specification: Part 8 – UA–Server		–
Compliance Part 9 UA Client	–	OPC Test Lab Specification: Part 9 – UA–Client		–

CONTENTS

European foreword 8

Annex ZA (normative) Normative references to international publications with their corresponding European publications..... 9

1 Scope 21

2 Normative references 21

3 Terms, definitions, and abbreviated terms 22

3.1 Terms and definitions 22

3.2 Abbreviated terms 23

4 Overview..... 23

4.1 General..... 23

4.2 ConformanceUnit..... 24

4.3 Profiles..... 24

4.4 Profile Categories 25

5 Conformance Units..... 25

5.1 Overview..... 25

5.2 Services..... 26

5.3 Transport and communication related features..... 35

5.4 Information Model and AddressSpace related features 44

5.5 Miscellaneous..... 62

6 Profiles..... 63

6.1 Overview..... 63

6.2 Profile list 63

6.3 Conventions for Profile definitions..... 70

6.4 Profile versioning 70

6.5 Applications 70

6.6 Profile tables..... 72

6.6.1 General..... 72

6.6.2 Core Server Facet..... 72

6.6.3 Core 2017 Server Facet 72

6.6.4 Sessionless Server Facet..... 73

6.6.5 Reverse Connect Server Facet..... 74

6.6.6 Base Server Behaviour Facet 74

6.6.7 Request State Change Server Facet 74

6.6.8 Subnet Discovery Server Facet..... 74

6.6.9 Global Certificate Management Server Facet..... 75

6.6.10 Authorization Service Server Facet..... 75

6.6.11 KeyCredential Service Server Facet 75

6.6.12 Attribute WriteMask Server Facet 75

6.6.13	File Access Server Facet	76
6.6.14	Documentation Server Facet	76
6.6.15	Embedded DataChange Subscription Server Facet.....	76
6.6.16	Standard DataChange Subscription Server Facet.....	77
6.6.17	Standard DataChange Subscription 2017 Server Facet.....	77
6.6.18	Enhanced DataChange Subscription Server Facet	77
6.6.19	Enhanced DataChange Subscription 2017 Server Facet.....	78
6.6.20	Durable Subscription Server Facet	78
6.6.21	Data Access Server Facet	78
6.6.22	ComplexType Server Facet	79
6.6.23	ComplexType 2017 Server Facet	79
6.6.24	Standard Event Subscription Server Facet.....	79
6.6.25	Address Space Notifier Server Facet.....	80
6.6.26	A & C Base Condition Server Facet.....	80
6.6.27	A & C Refresh2 Server Facet.....	81
6.6.28	A & C Address Space Instance Server Facet.....	81
6.6.29	A & C Enable Server Facet	81
6.6.30	A & C AlarmMetrics Server Facet	82
6.6.31	A & C Alarm Server Facet.....	82
6.6.32	A & C Acknowledgeable Alarm Server Facet	83
6.6.33	A & C Exclusive Alarming Server Facet.....	83
6.6.34	A & C Non-Exclusive Alarming Server Facet.....	83
6.6.35	A & C Previous Instances Server Facet.....	84
6.6.36	A & C Dialog Server Facet	84
6.6.37	A & C CertificateExpiration Server Facet.....	84
6.6.38	A & E Wrapper Facet	85
6.6.39	Method Server Facet.....	86
6.6.40	Auditing Server Facet.....	86
6.6.41	Node Management Server Facet.....	86
6.6.42	User Role Base Server Facet	87
6.6.43	User Role Management Server Facet	87
6.6.44	State Machine Server Facet.....	87
6.6.45	Client Redundancy Server Facet	87
6.6.46	Redundancy Transparent Server Facet.....	88
6.6.47	Redundancy Visible Server Facet.....	88
6.6.48	Historical Raw Data Server Facet.....	88
6.6.49	Historical Aggregate Server Facet	89
6.6.50	Historical Data AtTime Server Facet.....	90
6.6.51	Historical Access Modified Data Server Facet.....	90

6.6.52	Historical Annotation Server Facet.....	90
6.6.53	Historical Data Insert Server Facet	91
6.6.54	Historical Data Update Server Facet.....	91
6.6.55	Historical Data Replace Server Facet	91
6.6.56	Historical Data Delete Server Facet.....	92
6.6.57	Historical Access Structured Data Server Facet	92
6.6.58	Base Historical Event Server Facet.....	92
6.6.59	Historical Event Update Server Facet	92
6.6.60	Historical Event Replace Server Facet	93
6.6.61	Historical Event Insert Server Facet.....	93
6.6.62	Historical Event Delete Server Facet	93
6.6.63	Aggregate Subscription Server Facet	93
6.6.64	Nano Embedded Device Server Profile	95
6.6.65	Nano Embedded Device 2017 Server Profile	95
6.6.66	Micro Embedded Device Server Profile	95
6.6.67	Micro Embedded Device 2017 Server Profile	95
6.6.68	Embedded UA Server Profile	96
6.6.69	Embedded 2017 UA Server Profile	96
6.6.70	Standard UA Server Profile	96
6.6.71	Standard 2017 UA Server Profile	96
6.6.72	Core Client Facet.....	97
6.6.73	Core 2017 Client Facet	97
6.6.74	Sessionless Client Facet	98
6.6.75	Reverse Connect Client Facet	98
6.6.76	Base Client Behaviour Facet.....	98
6.6.77	Discovery Client Facet	99
6.6.78	Subnet Discovery Client Facet.....	99
6.6.79	Global Discovery Client Facet.....	99
6.6.80	Global Certificate Management Client Facet	99
6.6.81	KeyCredential Service Client Facet.....	100
6.6.82	Access Token Request Client Facet	100
6.6.83	AddressSpace Lookup Client Facet.....	100
6.6.84	Request State Change Client Facet.....	101
6.6.85	File Access Client Facet.....	101
6.6.86	Entry Level Support 2015 Client Facet	101
6.6.87	Multi-Server Client Connection Facet	102
6.6.88	Documentation – Client	102
6.6.89	Attribute Read Client Facet	102
6.6.90	Attribute Write Client Facet	103

6.6.91	DataChange Subscriber Client Facet	103
6.6.92	Durable Subscription Client Facet	104
6.6.93	DataAccess Client Facet.....	104
6.6.94	Event Subscriber Client Facet.....	105
6.6.95	Base Event Processing Client Facet.....	105
6.6.96	Notifier and Source Hierarchy Client Facet.....	105
6.6.97	A & C Base Condition Client Facet	106
6.6.98	A & C Refresh2 Client Facet.....	106
6.6.99	A & C Address Space Instance Client Facet	106
6.6.100	A & C Enable Client Facet	107
6.6.101	A & C AlarmMetrics Client Facet	107
6.6.102	A & C Alarm Client Facet	107
6.6.103	A & C Exclusive Alarming Client Facet.....	108
6.6.104	A & C Non-Exclusive Alarming Client Facet	108
6.6.105	A & C Previous Instances Client Facet.....	108
6.6.106	A & C Dialog Client Facet	109
6.6.107	A & C CertificateExpiration Client Facet	109
6.6.108	A & E Proxy Facet.....	109
6.6.109	Method Client Facet.....	111
6.6.110	Auditing Client Facet.....	111
6.6.111	Node Management Client Facet	111
6.6.112	Advanced Type Programming Client Facet	112
6.6.113	User Role Management Client Facet.....	112
6.6.114	State Machine Client Facet.....	112
6.6.115	Diagnostic Client Facet	113
6.6.116	Redundant Client Facet	113
6.6.117	Redundancy Switch Client Facet	113
6.6.118	Historical Access Client Facet.....	113
6.6.119	Historical Data AtTime Client Facet	114
6.6.120	Historical Aggregate Client Facet	114
6.6.121	Historical Annotation Client Facet.....	115
6.6.122	Historical Access Modified Data Client Facet	115
6.6.123	Historical Data Insert Client Facet	116
6.6.124	Historical Data Update Client Facet.....	116
6.6.125	Historical Data Replace Client Facet	116
6.6.126	Historical Data Delete Client Facet	116
6.6.127	Historical Access Client Server Timestamp Facet.....	117
6.6.128	Historical Structured Data Access Client Facet	117
6.6.129	Historical Structured Data AtTime Client Facet	117

6.6.130	Historical Structured Data Modified Client Facet	117
6.6.131	Historical Structured Data Insert Client Facet	117
6.6.132	Historical Structured Data Update Client Facet	118
6.6.133	Historical Structured Data Replace Client Facet.....	118
6.6.134	Historical Structured Data Delete Client Facet.....	118
6.6.135	Historical Events Client Facet	118
6.6.136	Historical Event Insert Client Facet	119
6.6.137	Historical Event Update Client Facet.....	119
6.6.138	Historical Event Replace Client Facet.....	119
6.6.139	Historical Event Delete Client Facet.....	119
6.6.140	Aggregate Subscriber Client Facet	120
6.6.141	Standard UA Client Profile	121
6.6.142	Standard UA Client 2017 Profile	121
6.6.143	UA-TCP UA-SC UA-Binary	122
6.6.144	HTTPS UA-Binary	122
6.6.145	HTTPS UA-XML	123
6.6.146	HTTPS UA-JSON	123
6.6.147	WSS UA-SC UA-Binary.....	123
6.6.148	WSS UA-JSON.....	124
6.6.149	Security User Access Control Full.....	124
6.6.150	Security User Access Control Base	124
6.6.151	Security Time Synchronization.....	125
6.6.152	Best Practice – Audit Events	125
6.6.153	Best Practice – Alarm Handling.....	125
6.6.154	Best Practice – Random Numbers	125
6.6.155	Best Practice – Timeouts	125
6.6.156	Best Practice – Administrative Access	126
6.6.157	Best Practice – Strict Message Handling	126
6.6.158	Best Practice – Audit Events Client.....	126
6.6.159	TransportSecurity – TLS 1.2	126
6.6.160	TransportSecurity – TLS 1.2 with PFS	127
6.6.161	SecurityPolicy – None	127
6.6.162	SecurityPolicy – Basic128Rsa15.....	127
6.6.163	SecurityPolicy – Basic256	128
6.6.164	SecurityPolicy [A] – Aes128-Sha256-RsaOaep	128
6.6.165	SecurityPolicy [B] – Basic256Sha256	128
6.6.166	SecurityPolicy – Aes256-Sha256-RsaPss	129
6.6.167	User Token – Anonymous Facet.....	129
6.6.168	User Token – User Name Password Server Facet	130

6.6.169	User Token – X509 Certificate Server Facet	130
6.6.170	User Token – Issued Token Server Facet	130
6.6.171	User Token – Issued Token Windows Server Facet	130
6.6.172	User Token – JWT Server Facet	131
6.6.173	User Token – User Name Password Client Facet	131
6.6.174	User Token – X509 Certificate Client Facet.....	131
6.6.175	User Token – Issued Token Client Facet.....	131
6.6.176	User Token – Issued Token Windows Client Facet.....	132
6.6.177	User Token – JWT Client Facet.....	132
6.6.178	Global Discovery Server Profile	132
6.6.179	Global Discovery Server 2017 Profile	132
6.6.180	Global Discovery and Certificate Management Server	133
6.6.181	Global Discovery and Certificate Mgmt 2017 Server	133
6.6.182	Global Certificate Management Client Profile	133
6.6.183	Global Certificate Management Client 2017 Profile	133
6.6.184	Global Service Authorization Request Server Facet	134
6.6.185	Global Service KeyCredential Pull Facet	134
6.6.186	Global Service KeyCredential Push Facet.....	134
	Bibliography	136
	Figure 1 – Profile – ConformanceUnit – TestCases.....	24
	Figure 2 – HMI Client sample	71
	Figure 3 – Embedded Server sample.....	71
	Figure 4 – Standard UA Server sample	72
	Table 1 – Profile Categories	25
	Table 2 – Conformance Groups	25
	Table 3 – Discovery Services	27
	Table 4 – Session Services	28
	Table 5 – Node Management Services	30
	Table 6 – View Services	30
	Table 7 – Attribute Services	31
	Table 8 – Method Services	32
	Table 9 – Monitored Item Services.....	32
	Table 10 – Subscription Services	34
	Table 11 – Security.....	36
	Table 12 – Protocol and Encoding	43
	Table 13 – Base Information	44
	Table 14 – Address Space Model	47

Table 15 – Data Access.....	48
Table 16 – Alarms and Conditions.....	49
Table 17 – Historical Access	52
Table 18 – Aggregates	55
Table 19 – Auditing.....	61
Table 20 – Redundancy.....	61
Table 21 – Global Discovery Server	61
Table 22 – Miscellaneous	62
Table 23 – Profile list	64
Table 24 – Core 2017 Server Facet	73
Table 25 – Sessionless Server Facet.....	73
Table 26 – Reverse Connect Server Facet	74
Table 27 – Base Server Behaviour Facet.....	74
Table 28 – Request State Change Server Facet.....	74
Table 29 – Subnet Discovery Server Facet.....	74
Table 30 – Global Certificate Management Server Facet	75
Table 31 – Authorization Service Server Facet	75
Table 32 – KeyCredential Service Server Facet	75
Table 33 – Attribute WriteMask Server Facet.....	76
Table 34 – File Access Server Facet.....	76
Table 35 – Documentation Server Facet.....	76
Table 36 – Embedded DataChange Subscription Server Facet	77
Table 37 – Standard DataChange Subscription 2017 Server Facet	77
Table 38 – Enhanced DataChange Subscription 2017 Server Facet.....	78
Table 39 – Durable Subscription Server Facet.....	78
Table 40 – Data Access Server Facet.....	79
Table 41 – ComplexType 2017 Server Facet.....	79
Table 42 – Standard Event Subscription Server Facet	80
Table 43 – Address Space Notifier Server Facet	80
Table 44 – A & C Base Condition Server Facet	81
Table 45 – A & C Refresh2 Server Facet	81
Table 46 – A & C Address Space Instance Server Facet.....	81
Table 47 – A & C Enable Server Facet	82
Table 48 – A & C AlarmMetrics Server Facet.....	82
Table 49 – A & C Alarm Server Facet	82
Table 50 – A & C Acknowledgeable Alarm Server Facet.....	83
Table 51 – A & C Exclusive Alarming Server Facet	83
Table 52 – A & C Non-Exclusive Alarming Server Facet	84
Table 53 – A & C Previous Instances Server Facet	84

Table 54 – A & C Dialog Server Facet	84
Table 55 – A & C CertificateExpiration Server Facet	85
Table 56 – A & E Wrapper Facet	85
Table 57 – Method Server Facet.....	86
Table 58 – Auditing Server Facet.....	86
Table 59 – Node Management Server Facet	86
Table 60 – User Role Base Server Facet.....	87
Table 61 – User Role Management Server Facet.....	87
Table 62 – State Machine Server Facet.....	87
Table 63 – Client Redundancy Server Facet	88
Table 64 – Redundancy Transparent Server Facet	88
Table 65 – Redundancy Visible Server Facet	88
Table 66 – Historical Raw Data Server Facet	88
Table 67 – Historical Aggregate Server Facet	89
Table 68 – Historical Data AtTime Server Facet.....	90
Table 69 – Historical Access Modified Data Server Facet.....	90
Table 70 – Historical Annotation Server Facet.....	91
Table 71 – Historical Data Insert Server Facet	91
Table 72 – Historical Data Update Server Facet.....	91
Table 73 – Historical Data Replace Server Facet	91
Table 74 – Historical Data Delete Server Facet.....	92
Table 75 – Historical Access Structured Data Server Facet.....	92
Table 76 – Base Historical Event Server Facet.....	92
Table 77 – Historical Event Update Server Facet	93
Table 78 – Historical Event Replace Server Facet.....	93
Table 79 – Historical Event Insert Server Facet.....	93
Table 80 – Historical Event Delete Server Facet	93
Table 81 – Aggregate Subscription Server Facet	94
Table 82 – Nano Embedded Device 2017 Server Profile	95
Table 83 – Micro Embedded Device 2017 Server Profile	96
Table 84 – Embedded 2017 UA Server Profile	96
Table 85 – Standard 2017 UA Server Profile	97
Table 86 – Core 2017 Client Facet	97
Table 87 – Sessionless Client Facet	98
Table 88 – Reverse Connect Client Facet	98
Table 89 – Base Client Behaviour Facet.....	98
Table 90 – Discovery Client Facet	99
Table 91 – Subnet Discovery Client Facet.....	99
Table 92 – Global Discovery Client Facet.....	99

Table 93 – Global Certificate Management Client Facet.....	100
Table 94 – KeyCredential Service Client Facet.....	100
Table 95 – Access Token Request Client Facet	100
Table 96 – AddressSpace Lookup Client Facet	101
Table 97 – Request State Change Client Facet.....	101
Table 98 – File Access Client Facet	101
Table 99 – Entry Level Support 2015 Client Facet.....	102
Table 100 – Multi-Server Client Connection Facet.....	102
Table 101 – Documentation – Client	102
Table 102 – Attribute Read Client Facet	103
Table 103 – Attribute Write Client Facet.....	103
Table 104 – DataChange Subscriber Client Facet.....	103
Table 105 – Durable Subscription Client Facet	104
Table 106 – DataAccess Client Facet	104
Table 107 – Event Subscriber Client Facet	105
Table 108 – Base Event Processing Client Facet	105
Table 109 – Notifier and Source Hierarchy Client Facet.....	106
Table 110 – A & C Base Condition Client Facet.....	106
Table 111 – A & C Refresh2 Client Facet	106
Table 112 – A & C Address Space Instance Client Facet.....	106
Table 113 – A & C Enable Client Facet.....	107
Table 114 – A & C AlarmMetrics Client Facet.....	107
Table 115 – A & C Alarm Client Facet.....	107
Table 116 – A & C Exclusive Alarming Client Facet.....	108
Table 117 – A & C Non-Exclusive Alarming Client Facet.....	108
Table 118 – A & C Previous Instances Client Facet.....	109
Table 119 – A & C Dialog Client Facet.....	109
Table 120 – A & C CertificateExpiration Client Facet.....	109
Table 121 – A & E Proxy Facet	110
Table 122 – Method Client Facet.....	111
Table 123 – Auditing Client Facet.....	111
Table 124 – Node Management Client Facet.....	111
Table 125 – Advanced Type Programming Client Facet.....	112
Table 126 – User Role Management Client Facet	112
Table 127 – State Machine Client Facet	112
Table 128 – Diagnostic Client Facet.....	113
Table 129 – Redundant Client Facet.....	113
Table 130 – Redundancy Switch Client Facet.....	113
Table 131 – Historical Access Client Facet	113

Table 132 – Historical Data AtTime Client Facet	114
Table 133 – Historical Aggregate Client Facet.....	114
Table 134 – Historical Annotation Client Facet	115
Table 135 – Historical Access Modified Data Client Facet	115
Table 136 – Historical Data Insert Client Facet.....	116
Table 137 – Historical Data Update Client Facet.....	116
Table 138 – Historical Data Replace Client Facet	116
Table 139 – Historical Data Delete Client Facet	116
Table 140 – Historical Access Client Server Timestamp Facet	117
Table 141 – Historical Structured Data Access Client Facet	117
Table 142 – Historical Structured Data AtTime Client Facet.....	117
Table 143 – Historical Structured Data Modified Client Facet	117
Table 144 – Historical Structured Data Insert Client Facet.....	118
Table 145 – Historical Structured Data Update Client Facet	118
Table 146 – Historical Structured Data Replace Client Facet.....	118
Table 147 – Historical Structured Data Delete Client Facet.....	118
Table 148 – Historical Events Client Facet	119
Table 149 – Historical Event Insert Client Facet	119
Table 150 – Historical Event Update Client Facet	119
Table 151 – Historical Event Replace Client Facet.....	119
Table 152 – Historical Event Delete Client Facet.....	120
Table 153 – Aggregate Subscriber Client Facet	120
Table 154 – Standard UA Client 2017 Profile	122
Table 155 – UA-TCP UA-SC UA-Binary	122
Table 156 – HTTPS UA-Binary	123
Table 157 – HTTPS UA-XML	123
Table 158 – HTTPS UA-JSON.....	123
Table 159 – WSS UA-SC UA-Binary.....	124
Table 160 – WSS UA-JSON.....	124
Table 161 – Security User Access Control Full.....	124
Table 162 – Security User Access Control Base	124
Table 163 – Security Time Synchronization.....	125
Table 164 – Best Practice – Audit Events	125
Table 165 – Best Practice – Alarm Handling.....	125
Table 166 – Best Practice – Random Numbers.....	125
Table 167 – Best Practice – Timeouts	126
Table 168 – Best Practice – Administrative Access.....	126
Table 169 – Best Practice – Strict Message Handling	126
Table 170 – Best Practice – Audit Events Client.....	126

Table 171 – TransportSecurity – TLS 1.2.....	127
Table 172 – TransportSecurity – TLS 1.2 with PFS	127
Table 173 – SecurityPolicy – None.....	127
Table 174 – SecurityPolicy [A] – Aes128-Sha256-RsaOaep	128
Table 175 – SecurityPolicy [B] – Basic256Sha256	129
Table 176 – SecurityPolicy – Aes256-Sha256-RsaPss.....	129
Table 177 – User Token – Anonymous Facet.....	130
Table 178 – User Token – User Name Password Server Facet	130
Table 179 – User Token – X509 Certificate Server Facet.....	130
Table 180 – User Token – Issued Token Server Facet.....	130
Table 181 – User Token – Issued Token Windows Server Facet.....	131
Table 182 – User Token – JWT Server Facet.....	131
Table 183 – User Token – User Name Password Client Facet.....	131
Table 184 – User Token – X509 Certificate Client Facet	131
Table 185 – User Token – Issued Token Client Facet	132
Table 186 – User Token – Issued Token Windows Client Facet	132
Table 187 – User Token – JWT Client Facet	132
Table 188 – Global Discovery Server 2017 Profile.....	133
Table 189 – Global Discovery and Certificate Mgmt 2017 Server	133
Table 190 – Global Certificate Management Client 2017 Profile	134
Table 191 – Global Service Authorization Request Server Facet.....	134
Table 192 – Global Service KeyCredential Pull Facet.....	134
Table 193 – Global Service KeyCredential Push Facet	135

OPC UNIFIED ARCHITECTURE –

Part 7: Profiles

1 Scope

This part of IEC 62541 defines 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 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 are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 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 62541-11, *OPC Unified Architecture – Part 11: Historical Access*

IEC 62541-12, *OPC Unified Architecture – Part 12: Discovery and Global Services*

IEC 62541-13, *OPC Unified Architecture – Part 13: Aggregates*

Compliance Part 8 UA Server: OPC Test Lab Specification: Part 8 – UA Server
<http://www.opcfoundation.org/Test/Part8/>

Compliance Part 9 UA Client: OPC Test Lab Specification: Part 9 – UA Client
<http://www.opcfoundation.org/Test/Part9/>

3 Terms, definitions and abbreviated terms

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 and the following apply.

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

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

NOTE An overview of the terms defined in this document 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.

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

category which 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 shall 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 Abbreviated terms

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 IEC 62541 series, which is the OPC Unified architecture 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 document. This relationship is also displayed by the OPC UA Compliance Test Tool.

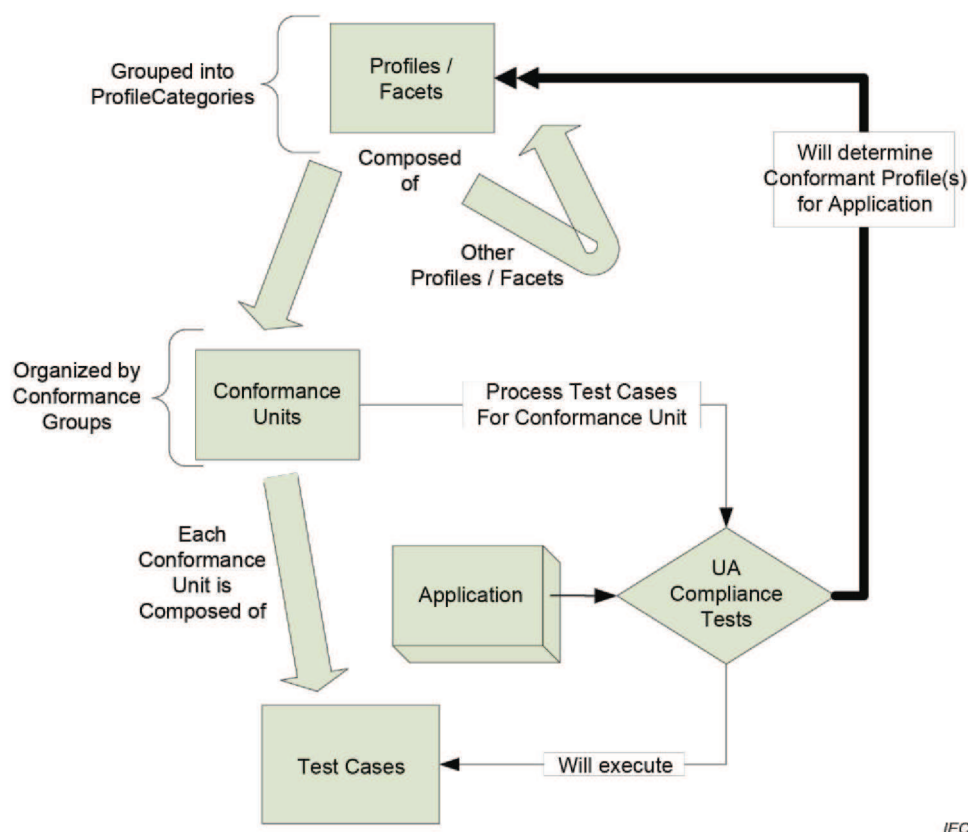


Figure 1 – Profile – ConformanceUnit – TestCases

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 OPC UA. 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 – Profile Categories

Category	Description
Client	Profiles of this category specify functions of an OPC UA Client.
Global Directory Service	Profiles of this category specify functions for global discovery and certificate management.
Security	Profiles 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. Profiles of this category apply to Clients and Servers.
Server	Profiles of this category specify functions of an OPC UA Server. The URI of such Profiles can be exposed in the Server capabilities.
Transport	Profiles of this category specify specific protocol mappings. The URI of such Profiles has to be part of an Endpoint Description. These Profiles apply to Clients and Servers.

5 Conformance Units

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 – Conformance Groups

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 Aggregates, including individual <i>ConformanceUnits</i> for each supported Aggregate as described in IEC 62541-13.
<i>Alarms and Conditions</i>	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.
<i>Attribute Services</i>	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.
<i>Discovery Services</i>	<i>ConformanceUnits</i> which focus on <i>Server</i> Endpoint <i>Discovery</i> .
GDS	<i>ConformanceUnits</i> for a GDS. Includes units for global discovery and global certificate management.

Group	Description
Historical Access	Access to archived data of <i>Node Attribute</i> values or Events.
<i>Method Services</i>	Methods represent the function calls of <i>Objects</i> . Methods 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 <i>Services</i>	<i>Clients</i> define <i>MonitoredItems</i> to subscribe to data and Events. Each <i>MonitoredItem</i> identifies the item to be monitored and the <i>Subscription</i> to use to send <i>Notifications</i> .
<i>Node Management Services</i>	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.
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.
<i>Session Services</i>	An (OPC UA) <i>Session</i> is an application layer connection.
<i>Subscription Services</i>	Subscriptions are used to report <i>Notifications</i> to the <i>Client</i> .
<i>View Services</i>	<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

Table 3 to Table 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*. In some cases, a *ConformanceUnit* will be derived from another *ConformanceUnit*. This parent unit will then be specified in the description of each derived unit. In such cases, the derived units inherit all of the tests of 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
Server	<i>Discovery</i> 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	<i>Discovery</i> Get Endpoints SessionLess	Support at least one endpoint for issueing SessionLess <i>Services</i> . Support obtaining such endpoints by accepting the Transport URI as a filter to the GetEndpoints <i>Service</i> with the query string "SL" appended to the Transport URI. E.g. "http://opcfoundation.org/UA-Profile/Transport/https-uajson?SL"
Server	<i>Discovery</i> Find Servers Self	Support the FindServers <i>Service</i> only for itself.
Server	<i>Discovery</i> 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	<i>Discovery</i> Register2	Call the RegisterServer2 <i>Service</i> to register with an external <i>Discovery Service</i> via a Secure Channel with a SecurityMode other than "None". This includes passing a list of short capability identifiers. The identifiers and their use are specified in IEC 62541-12.
Server	<i>Discovery</i> Server Announcement using mDNS	Provide mDNS functionality to announce a <i>Server</i> with its capabilities. The capability identifiers and the use of mDNS records for the purpose of OPC UA <i>Discovery</i> is specified in IEC 62541-12. Note that this functionality is only required for <i>Servers</i> that do not register with an LDS. The capability identifiers and their use in mDNS records are specified in IEC 62541-12.
Server	<i>Discovery</i> 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	<i>Discovery Client</i> Find Servers Basic	Uses the FindServers <i>Service</i> to obtain all <i>Servers</i> installed on a given platform.
Client	<i>Discovery Client</i> Find Servers with URI	Use FindServers <i>Service</i> to obtain URLs for specific <i>Server</i> URIs.
Client	<i>Discovery Client</i> Find Servers Dynamic	Detect new <i>Servers</i> after an initial FindServers <i>Service</i> call.
Client	<i>Discovery Client</i> Find Servers on Network	Support one of the options to locate <i>Servers</i> on the network.
Client	<i>Discovery Client</i> Find Servers on Network using LDS-ME	Use FindServersOnNetwork <i>Service</i> to obtain URLs for specific <i>Server</i> URIs. Note that this <i>Service</i> is available via the Local <i>Discovery Server</i> with multicast extension (LDS-ME).
Client	<i>Discovery Client</i> Find Servers on Network using mDNS	Use mDNS based <i>Service Discovery</i> to locate <i>Servers</i> on the same multicast network. The contents of mDNS records for OPC UA <i>Discovery</i> are described in IEC 62541-12. Note that this functionality is only required for <i>Clients</i> when there is no Local <i>Discovery Server</i> with multicast extension (LDS-ME). The capability identifiers and their use in mDNS records are specified in IEC 62541-12.
Client	<i>Discovery Client</i> Find Servers in GDS	Use the QueryServers <i>Method</i> on the GDS Directory <i>Object</i> to locate <i>Servers</i> that meet filter criteria specified in the request. This <i>Method</i> is specified in IEC 62541-12.

Category	Title	Description
<i>Client</i>	<i>Discovery Client</i> Find Applications in GDS	Use the QueryApplications <i>Method</i> on the GDS Directory <i>Object</i> to locate Applications that meet filter criteria specified in the request. This <i>Method</i> is specified in IEC 62541-12.
<i>Client</i>	<i>Discovery Client</i> Get Endpoints Basic	Uses the GetEndpoints <i>Service</i> to obtain all Endpoints for a given <i>Server</i> URI.
<i>Client</i>	<i>Discovery Client</i> Get Endpoints SessionLess	Uses the GetEndpoints <i>Service</i> with a filter to obtain Endpoints that can be used for SessionLess <i>Service</i> invocation. The filter is the Transport URI extended with the query string "SL". E.g. http://opcfoundation.org/UA-Profile/Transport/https-uajson?SL.
<i>Client</i>	<i>Discovery Client</i> Get Endpoints Dynamic	Detect changes to the Endpoints after an initial GetEndpoints <i>Service</i> call.
<i>Client</i>	<i>Discovery Client</i> Configure Endpoint	Allow specification of an Endpoint without going through the <i>Discovery Service</i> Set.

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
<i>Server</i>	<i>Session</i> General <i>Service</i> 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
<i>Server</i>	<i>Session</i> Base	Support the <i>Session Service</i> Set (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.
<i>Server</i>	<i>Session</i> Change User	Support the use of ActivateSession to change the <i>Session</i> user.
<i>Server</i>	<i>Session</i> Cancel	Support the Cancel <i>Service</i> to cancel outstanding requests.
<i>Server</i>	<i>Session</i> Minimum 1	Support minimum 1 <i>Session</i> (total).
<i>Server</i>	<i>Session</i> Minimum 2 Parallel	Support minimum 2 parallel <i>Sessions</i> (total for all <i>Clients</i>).
<i>Server</i>	<i>Session</i> Minimum 50 Parallel	Support minimum 50 parallel <i>Sessions</i> (total for all <i>Clients</i>).
<i>Server</i>	<i>Session</i> Sessionless Invocation	Defines the support of the SessionlessInvoke <i>Service</i> defined in IEC 62541-4 to process any of the <i>Services</i> (like Read/Write, Browse, or Call) that are designated for Sessionless Invocation.

Category	Title	Description
<i>Client</i>	<i>Session Client</i> General Service Behaviour	Implement basic <i>Service</i> behaviour. This includes in particular: <ul style="list-style-type: none"> — 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
<i>Client</i>	<i>Session Client</i> Base	Use the <i>Session Service</i> Set (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: <ol style="list-style-type: none"> 1) The Application <i>Certificate</i> and Nonce are optional. 2) The signatures are null/empty.
<i>Client</i>	<i>Session Client</i> 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.
<i>Client</i>	<i>Session Client</i> 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.
<i>Client</i>	<i>Session Client</i> Impersonate	Uses ActivateSession to change the Session user (impersonation).
<i>Client</i>	<i>Session Client</i> KeepAlive	Make periodic requests to keep the <i>Session</i> alive.
<i>Client</i>	<i>Session Client</i> Detect Shutdown	Read or monitor the ServerStatus/State Variable to recognize a potential shutdown of the Server and clean up resources.
<i>Client</i>	<i>Session Client</i> Cancel	Use the Cancel <i>Service</i> to cancel outstanding requests.
<i>Client</i>	<i>Session Client</i> Auto Reconnect	Automatic <i>Client</i> reconnect including: <ul style="list-style-type: none"> — 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
<i>Client</i>	<i>Session Client</i> Single Session	The <i>Client</i> shall interoperate with <i>Servers</i> that only support one <i>Session</i> .
<i>Client</i>	<i>Session Client</i> SessionLess Service Calls	Defines the use of the SessionlessInvoke <i>Service</i> defined in IEC 62541-4 to request one of the <i>Services</i> (like Read or Browse) that are allowed for Sessionless Invocation. IEC 62541-6 specifies which transports may be used and how.

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
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
Server	View Basic	Support the View <i>Service Set</i> (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 Remote Nodes Browse	The <i>Client</i> can browse to nodes that have an extended NodeID that reference a <i>Server</i> different than the originating <i>Server</i> . This includes automatic connection to the remote <i>Server</i> . It is acceptable that the <i>Server</i> configuration information be pre-configured on the <i>Client</i> and / or that the user is prompted to connect.
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 Remote Nodes Translate Browse	The <i>Client</i> can translate browse paths that include nodes with extended NodeID that reference a <i>Server</i> different than the originating <i>Server</i> and return them as part of the TranslateBrowsePathsToNodeIds <i>Service</i> . It is acceptable that the <i>Server</i> configuration information be pre-configured on the <i>Client</i> .
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
Server	<i>Attribute Read</i>	Supports the <i>Read 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	<i>Attribute Read Complex</i>	Supports reading and encoding Values with structured DataTypes.
Server	<i>Attribute Write Values</i>	Supports writing to values to one or more <i>Attributes</i> of one or more <i>Nodes</i> .
Server	<i>Attribute Write Complex</i>	Supports writing and decoding Values with structured DataTypes.
Server	<i>Attribute Write StatusCode & Timestamp</i>	Supports writing of <i>StatusCode</i> and <i>Timestamps</i> along with the <i>Value</i> .
Server	<i>Attribute Write Index</i>	Supports the <i>IndexRange</i> to write a single element or a range of elements when the <i>Attribute</i> value is an array and partial updates are allowed for this array.
Server	<i>Attribute Alternate Encoding</i>	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	<i>Attribute Historical Read</i>	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> shall be supported.
Server	<i>Attribute Historical Update</i>	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 shall 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	<i>Attribute Client Read Base</i>	Use the <i>Read 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	<i>Attribute Client Remote Nodes Attribute Access</i>	The <i>Client</i> can retrieve attributes of nodes that have an extended <i>NodeID</i> that reference a <i>Server</i> different than the originating <i>Server</i> . This requires a connection to the remote <i>Server</i> for access (not necessarily displayed as a connection). It is acceptable that the <i>Server</i> configuration information be pre-configured on the <i>Client</i> .
Client	<i>Attribute Client Read with proper Encoding</i>	This <i>ConformanceUnit</i> refers to the ability of a <i>Client</i> to discover the available encodings and choose a specific one when calling the <i>Read Service</i> .
Client	<i>Attribute Client Read Complex</i>	Read and decode Values with structured DataTypes.
Client	<i>Attribute Client Write Base</i>	Use the <i>Write 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	<i>Attribute Client Write Complex</i>	Write and Encode Values with structured DataTypes.

Category	Title	Description
<i>Client</i>	<i>Attribute Client</i> Write Quality & Timestamp	Use the Write <i>Service</i> to also write StatusCode and/or Timestamps along with a Value.
<i>Client</i>	<i>Attribute Client</i> Historical Read	The <i>Client</i> makes use of the HistoryRead 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 ReadRaw, ReadAtTime, ReadProcessed, ReadModified or ReadEvents.
<i>Client</i>	<i>Attribute Client</i> Historical Updates	The <i>Client</i> makes use of the HistoryUpdate service. The details of this usage are provided by additional <i>ConformanceUnits</i> , but at least one or more of the following shall be provided InsertData, InsertEvents, ReplaceData, ReplaceEvents, UpdateData, UpdateEvents, DeleteData or DeleteEvents or DeleteAtTime.

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
<i>Server</i>	<i>Method</i> Call	Support the Call <i>Service</i> to call (invoke) a <i>Method</i> which includes support for <i>Method</i> Parameters.
<i>Client</i>	<i>Method Client</i> 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
<i>Server</i>	Monitor Basic	Support the following <i>MonitoredItem Services</i> : CreateMonitoredItems, ModifyMonitoredItems, DeleteMonitoredItems and SetMonitoringMode.
<i>Server</i>	Monitor Value Change	Support creation of <i>MonitoredItems</i> for <i>Attribute</i> value changes. This includes support of the IndexRange to select a single element or a range of elements when the <i>Attribute</i> value is an array.
<i>Server</i>	Monitor Complex Value	Supports monitoring and encoding Values with structured DataTypes.
<i>Server</i>	Monitored Items Deadband Filter	Supports an absolute Deadband filter as a DataChangeFilter for numeric data types.
<i>Server</i>	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> .
<i>Server</i>	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 Stack <i>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.
<i>Server</i>	Monitor Items 2	Support at least 2 <i>MonitoredItems</i> per <i>Subscription</i> where the size of each <i>MonitoredItem</i> is at least equal to size of Double.

Category	Title	Description
Server	Monitor Items 10	Support at least 10 <i>MonitoredItems</i> per <i>Subscription</i> where the size of each <i>MonitoredItem</i> is at least equal to size of Double.
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 Subscriptions for half of the required Sessions.
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 Subscriptions for half of the required Sessions.
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> .
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 <i>Client</i> 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 the 'TypeOf' complex <i>Event</i> filter operator.
Client	Monitor Client Value Change	Use the <i>MonitoredItem Service</i> Set 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 Complex Value	Monitor and decode Values with structured DataTypes.
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</i> Set to create <i>MonitoredItems</i> for <i>Event</i> notifications.

Category	Title	Description
<i>Client</i>	Monitor <i>Client</i> Event Filter	Use the <i>Event</i> filter when calling <i>CreateMonitoredItems</i> to filter the desired Events and to select the columns to be provided for each <i>Event Notification</i> .
<i>Client</i>	Monitor <i>Client</i> Complex Event Filter	Use of the 'TypeOf' complex <i>Event</i> filter operator.
<i>Client</i>	Monitor <i>Client</i> Modify	Use <i>ModifyMonitoredItems Service</i> to change the configuration setting. Use <i>SetMonitoringMode Service</i> to disable/enable sampling and/or publishing.
<i>Client</i>	Monitor <i>Client</i> 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.

Table 10 – Subscription Services

Category	Title	Description
<i>Server</i>	<i>Subscription</i> Basic	Support the following <i>Subscription Services</i> : <i>CreateSubscription</i> , <i>ModifySubscription</i> , <i>DeleteSubscriptions</i> , <i>Publish</i> , <i>Republish</i> and <i>SetPublishingMode</i> .
<i>Server</i>	<i>Subscription</i> Minimum 1	Support at least 1 <i>Subscription</i> per <i>Session</i> . This number has to be supported for all of the minimum required sessions.
<i>Server</i>	<i>Subscription</i> Minimum 02	Support at least 2 <i>Subscriptions</i> per <i>Session</i> . This number has to be supported for at least half of the minimum required sessions.
<i>Server</i>	<i>Subscription</i> Minimum 05	Support at least 5 <i>Subscriptions</i> per <i>Session</i> . This number has to be supported for at least half of the minimum required sessions.
<i>Server</i>	<i>Subscription</i> Publish Min 02	Support at least 2 <i>Publish Service</i> requests per <i>Session</i> . This number has to be supported for all of the minimum required sessions. Support of a <i>NotificationMessage</i> retransmission queue is not required; if not available, the <i>Republish Service</i> returns <i>Bad_MessageNotAvailable</i> .
<i>Server</i>	<i>Subscription</i> Publish Min 05	Support at least 5 <i>Publish 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 <i>Publish</i> requests per session as the size of the <i>NotificationMessage</i> retransmission queue for <i>Republish</i> .
<i>Server</i>	<i>Subscription</i> Publish Min 10	Support at least 10 <i>Publish 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 <i>Publish</i> requests per session as the size of the <i>NotificationMessage</i> retransmission queue for <i>Republish</i> .
<i>Server</i>	<i>Subscription</i> Publish Discard Policy	Respect the specified policy for discarding <i>Publish Service</i> requests. If the maximum number of <i>Publish Service</i> requests has been queued and a new <i>Publish Service</i> request arrives, the "oldest" <i>Publish</i> request has to be discarded by returning the proper error.

Category	Title	Description
Server	Subscription Transfer	Support TransferSubscriptions Service to transfer a Subscription from one Session to another.
Server	Subscription Durable	Support setting Subscriptions in durable mode. This mode requires that collected data and events are stored and delivered even if a Client was disconnected for a longer time or the Server was restarted. Support one of the "Subscription Durable StorageLevel nnn" ConformanceUnits.
Server	Subscription Durable StorageLevel Small	Support at least 20 monitored items with a queue size of 10 000 for each item and where the size of each MonitoredItem is at least equal to size of Double. This requires storage capacity for 200 000 values of DataType Double.
Server	Subscription Durable StorageLevel Medium	Support at least 100 monitored items with a queue size of 50 000 for each item and where the size of each MonitoredItem is at least equal to size of Double. This requires storage capacity for 5 million values of DataType Double.
Server	Subscription Durable StorageLevel High	Support at least 2 000 monitored items with a queue size of 200 000 for each item and where the size of each MonitoredItem is at least equal to size of Double. This requires storage capacity for 400 million values of DataType Double.
Client	Subscription Client Basic	Use the Subscription and MonitoredItem Service Set as an efficient means to detect changes of Attribute values and / or to receive Event occurrences. Set appropriate intervals for publishing, keep alive notifications and total Subscription lifetime. Supply a sufficient number of Publish requests to the Server so that Notifications can be sent whenever a publish timer expires. Acknowledge received Notifications with subsequent Publish requests.
Client	Subscription Client Fallback	The Client shall interoperate with Servers that do not support Subscriptions, or have exhausted Subscription limits, for Monitoring by using Read Service.
Client	Subscription Client Republish	Evaluate the sequence number in Notifications to detect lost Notifications. Use Republish to request missing Notifications.
Client	Subscription Client Modify	Allow modification of the Subscription configuration using the ModifySubscription Service.
Client	Subscription Client TransferSubscriptions	The Client supports transferring Subscription from other Clients. This ConformanceUnit is used as part of redundant Clients.
Client	Subscription Client Multiple	Use multiple Subscriptions to reduce the payload of individual Notifications.
Client	Subscription Client Publish Configurable	Send multiple Publish Service requests to assure that the Server is always able to send Notifications. The number of parallel Publish Service requests per Session shall be configurable.
Client	Subscription Client Durable	Use durable Subscriptions.

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
Security	Security User Name Password	The <i>Server</i> supports User Name/Password combination(s). The token will be encrypted if required by the security policy of the User Token Policy or by the security policy of the endpoint. An unencrypted token either requires message encryption or means outside the scope of OPC UA to secure the identity token so that it cannot be retrieved by sniffing the communication. One option would be a secure transport like a VPN.
Security	Security User X509	The <i>Server</i> supports a public/private key pair for user identity. The use of this feature shall be able to be enabled or disabled by an administrator.
Security	Security User IssuedToken Kerberos	<p>The <i>Server</i> supports a Kerberos <i>Server</i> token for User Identity. The use of this feature shall be able to be enabled or disabled by an Administrator. The use of this token is defined in Kerberos Token Documentation.</p> <p>The token will be encrypted if required by the security policy of the User Token Policy or by the security policy of the endpoint. An unencrypted token either requires message encryption or means outside the scope of OPC UA to secure the identity token so that it cannot be retrieved by sniffing the communication. One option would be a secure transport like a VPN.</p>
Security	Security User IssuedToken Kerberos Windows	<p>The <i>Server</i> supports the Windows implementation of Kerberos Tokens. This <i>ConformanceUnit</i> only applies if the "Security User IssuedToken Kerberos" is supported.</p> <p>The token will be encrypted if required by the security policy of the User Token Policy or by the security policy of the endpoint. An unencrypted token either requires message encryption or means outside the scope of OPC UA to secure the identity token so that it cannot be retrieved by sniffing the communication. One option would be a secure transport like a VPN.</p>
Security	Security User JWT IssuedToken	<p>The <i>Server</i> supports a JSON Web Token (JWT) for user identity. IEC 62541-6 describes OAuth2 and JWTs in more detail. The use of this feature shall be able to be enabled or disabled by an administrator.</p> <p>The token will be encrypted if required by the security policy of the User Token Policy or by the security policy of the endpoint. An unencrypted token either requires message encryption or means outside the scope of OPC UA to secure the identity token so that it cannot be retrieved by sniffing the communication. One option would be a secure transport like a VPN.</p>
Security	Security User Anonymous	The <i>Server</i> provides support for Anonymous access. The use of this feature shall be able to be enabled or disabled by an Administrator. By default, Anonymous access shall be disabled.
Security	Security User Name Password <i>Client</i>	<p>A <i>Client</i> uses a User Name/Password combination.</p> <p>The token will be encrypted if required by the security policy of the User Token Policy or by the security policy of the endpoint. An unencrypted token either requires message encryption or means outside the scope of OPC UA to secure the identity token so that it cannot be retrieved by sniffing the communication. One option would be a secure transport like a VPN.</p>
Security	Security User X509 <i>Client</i>	A <i>Client</i> uses a public/private key pair for user identity. This includes all validation and trust issues associated with a certificate.
Security	Security User IssuedToken Kerberos <i>Client</i>	<p>A <i>Client</i> uses a Kerberos <i>Server</i> token. The use of this token is defined by the Kerberos documentation.</p> <p>The token will be encrypted if required by the security policy of the User Token Policy or by the security policy of the endpoint. An unencrypted token either requires message encryption or means outside the scope of OPC UA to secure the identity token so that it cannot be retrieved by sniffing the communication. One option would be a secure transport like a VPN.</p>

Category	Title	Description
Security	Security User IssuedToken Kerberos Windows <i>Client</i>	<p>A <i>Client</i> uses the Windows implementation of Kerberos tokens. This <i>ConformanceUnit</i> only applies if the "Security User IssuedToken Kerberos <i>Client</i>" is supported.</p> <p>The token will be encrypted if required by the security policy of the User Token Policy or by the security policy of the endpoint. An unencrypted token either requires message encryption or means outside the scope of OPC UA to secure the identity token so that it cannot be retrieved by sniffing the communication. One option would be a secure transport like a VPN.</p>
Security	Security User JWT IssuedToken <i>Client</i>	<p>A <i>Client</i> uses a JSON Web Token (JWT) for user identity. IEC 62541-6 describes OAuth2 and JWTs in more detail.</p> <p>The token will be encrypted if required by the security policy of the User Token Policy or by the security policy of the endpoint. An unencrypted token either requires message encryption or means outside the scope of OPC UA to secure the identity token so that it cannot be retrieved by sniffing the communication. One option would be a secure transport like a VPN.</p>
Security	Security Invalid user token	<p><i>Servers</i> shall take proper measures to protect against attacks on user identity tokens. Such an attack is assumed if repeated connection attempts with invalid user identity tokens happen. See <i>ActivateSession Service</i> in IEC 62541-4.</p>
Security	Security User JWT Token Policy	<p>The Server supports one or more Endpoints with a UserTokenPolicy that includes a JWT IssuerEndpointUrl as defined in IEC 62541-6.</p> <p>For JWT the issuerEndpointUrl is a JSON object that includes all parameters that define the AuthorizationService.</p> <p>As part of the JWT Token Policy, the Server shall support at least one of the following Authority Profile Conformance Units. The URIs defined in the ConformanceUnit shall be exposed in the authorityProfileURI field of the JWT Token Policy.</p>
Security	Security User JWT Token Policy <i>Client</i>	<p>The Client understands and uses the Authorization Service definition inside the JWT UserTokenPolicy returned with GetEndpoints.</p> <p>It shall support at least one of the following Authority Profile Conformance Units. The URIs defined in the ConformanceUnit are in the authorityProfileURI field of the JWT Token Policy exposed in Server Endpoints.</p>
Security	OAuth2 Authority <i>Profile</i>	<p>This unit indicates support of OAuth2 over HTTPS to request access tokens.</p> <p>The URI for the interactions with this authority is "http://opcfoundation.org/UA/Authorization#OAuth2"</p>
Security	OPC UA Authority <i>Profile</i>	<p>This unit indicates support of the OPC UA <i>Methods</i> defined in IEC 62541-12 to request access tokens.</p> <p>The URI for the interactions with this authority is "http://opcfoundation.org/UA/Authorization#OPCUA"</p>
Security	Azure Identity Provider Authority <i>Profile</i>	<p>This unit indicates support of the Azure identity provider to request access tokens.</p> <p>The URI for the interactions with this authority is "http://opcfoundation.org/UA/Authorization#Azure"</p>
Security	Security <i>Certificate</i> Validation	<p>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.</p>
Security	Security Default ApplicationInstance Certificate	<p>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.</p>

Category	Title	Description
Security	Security – No Application Authentication	The <i>Server</i> supports being able to be configured for no application authentication, just User authentication and normal encryption/signing: <ul style="list-style-type: none"> – Configure <i>Server</i> to accept all certificates – <i>Certificates</i> are just used for message security (signing and encryption) – Users level is used for authentication
Security	Security Policy Required	Support at least Security Policy [A] and Security Policy [B]. Support of multiple Security Policies – even obsolete ones – is recommended. This will provide best interoperability and allows the end user to choose the required level of security. Obsolete Security Policies shall not be enabled/usable without administrative intervention.
Security	Security None CreateSession ActivateSession	When SecurityPolicy = None, the CreateSession and ActivateSession service allow for a NULL/empty signature and do not require Application <i>Certificates</i> or a Nonce.
Security	Security None CreateSession ActivateSession 1.0	The <i>Client</i> can connect to <i>Servers</i> that require a certificate being passed on Session establishment. The <i>Client</i> in this case will first try without a <i>Certificate</i> and if this fails present a <i>Certificate</i> .
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_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 TLS_DHE_RSA with AES_nnn_CBC_SHA256	The connection is established using TLS_DHE_RSA with AES_128_CBC_SHA256 or TLS_DHE_RSA with AES_256_CBC_SHA256. That has a MinAsymmetricKeyLength – 2048, MaxAsymmetricKeyLength – 4096, CertificateSignatureAlgorithm – RSA_SHA256. (TLS 1.2). <i>Clients</i> and <i>Servers</i> have to support both algorithms.
Security	Security Encryption Required	Encryption is required using the algorithms provided in the security algorithm suite.
Security	Security Signing Required	Signing is required using the algorithms provided 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.

Category	Title	Description
Security	Security Administration	<p>Allow configuration of the following Security related items (when they apply).</p> <ul style="list-style-type: none"> * select the allowed User identification policy or policies (e.g. Username/Password or X509). * 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 Certificates. * allow accepting any valid Certificate
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 <i>Certificate</i> Administration	Allow a site administrator to be able to assign a site specific ApplicationInstanceCertificate and if desired to configure a site-specific <i>Certificate</i> Authority (CA).
Security	Security Role <i>Server</i> Base	Support the User Authorization Information Model defined in UA Parts 3 and 5 – like Roles – and the RolePermissions and UserRolePermissions <i>Attributes</i> .
Security	Security Role Well Known	Support the well-known Roles "ConfigureAdmin" and "SecurityAdmin" with suggested permissions defined in IEC 62541-3.
Security	Security Role <i>Server</i> IdentityManagement	Allow authorized users to add and/or remove Identities from Roles with the appropriate <i>Methods</i> .
Security	Security Role <i>Server</i> Management	Allow authorized users to create new Roles and/or remove Roles with the appropriate <i>Methods</i> .
Security	Security Role <i>Server</i> Restrict Applications	Support adding applications to a Role with the appropriate <i>Methods</i> so that only these applications can use this Role.
Security	Security Role <i>Server</i> Restrict Endpoints	Support adding Endpoints to a Role with the appropriate <i>Methods</i> . With this restriction a Role is only applied when a <i>Client</i> connects via one of these Endpoints.
Security	Security Role <i>Server</i> DefaultRolePermissions	Allow authorized users to set the DefaultRolePermissions <i>Property</i> for certain NameSpaces. DefaultRolePermissions are applied if no RolePermissions are associated with a <i>Node</i> .
Security	Security Role <i>Server</i> RolePermissions	Allow authorized users to set the RolePermissions <i>Attribute</i> on <i>Nodes</i> .
Security	Security Role <i>Server</i> Authorization	Restrict access based on the configured Roles and permissions.
Security	Security Role <i>Client</i> Base	Understand and use the User Authorization Information Model defined in IEC 62541-5 and the RolePermissions <i>Attribute</i> .
Security	Security Role <i>Client</i> Management	Support creating new Roles and adding Identities as well as remove Roles or Identities using the appropriate <i>Methods</i> .
Security	Security Role <i>Client</i> Restrict Applications	Use the appropriate <i>Methods</i> to add applications to a Role so that only these applications can use this Role.
Security	Security Role <i>Client</i> Restrict Endpoints	Use the appropriate <i>Methods</i> to add Endpoints to a Role. With this restriction a Role is only applied when a <i>Client</i> connects via one of these Endpoints.
Security	Security Role <i>Client</i> DefaultRolePermissions	Ability to set the DefaultRolePermissions <i>Property</i> for certain NameSpaces. DefaultRolePermissions are applied if no RolePermissions are associated with a <i>Node</i> .
Security	Security Role <i>Client</i> RolePermissions	Support setting the RolePermissions <i>Attribute</i> on <i>Nodes</i> .

Category	Title	Description
Security	Pull Model for Global <i>Certificate</i> and TrustList Management	Use the <i>Certificate Management Services</i> of IEC 62541-12 for the Pull model to manage Application Instance <i>Certificates</i> and Trust Lists including Revocation Lists.
Security	Push Model for Global <i>Certificate</i> and TrustList Management	Support the <i>Certificate Management Services</i> of IEC 62541-12 for the Push model to manage Application Instance <i>Certificates</i> and Trust Lists including Revocation Lists.
Security	Pull Model for KeyCredential <i>Service</i>	Use the <i>Methods</i> on an instance of the KeyCredentialServiceType (Pull model) to obtain KeyCredentials as specified in IEC 62541-12.
Security	Push Model for KeyCredential <i>Service</i>	Support the KeyCredential <i>Services</i> Push model of IEC 62541-12 to obtain KeyCredentials. This includes support of one or more instances of the KeyCredentialConfigurationType and the <i>Methods</i> to update or delete credentials.
Security	Authorization Service Configuration Server	Support the Object Types defined in IEC 62541-12 to allow configuration of information needed to accept Access Tokens when presented by the Client during session establishment. Access Tokens are issued by Authorization Services.
Security	Authorization Service Client	Use the RequestAccessToken Method defined in IEC 62541-12.
Security	SymmetricSignatureAlgorithm_None	This algorithm does not apply.
Security	SymmetricSignatureAlgorithm_HMAC-SHA1	A keyed hash which is defined in https://tools.ietf.org/html/rfc2104 . The hash algorithm is SHA1 and is described in https://tools.ietf.org/html/rfc3174 . The URI is http://www.w3.org/2000/09/xmlsig#hmac-sha1 . No known exploits exist when using SHA1 with a keyed hash, however, SHA1 was broken in 2017 so use of this algorithm is not recommended.
Security	SymmetricSignatureAlgorithm_HMAC-SHA2-256	A keyed hash used for message authentication which is defined in https://tools.ietf.org/html/rfc2104 . The hash algorithm is SHA2 with 256 bits and described in https://tools.ietf.org/html/rfc4634 .
Security	SymmetricEncryptionAlgorithm_None	This algorithm does not apply.
Security	SymmetricEncryptionAlgorithm_AES128-CBC	The AES encryption algorithm which is defined in http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf . Multiple blocks encrypted using the CBC mode described in http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf . The key size is 128 bits. The block size is 16 bytes. The URI is http://www.w3.org/2001/04/xmlenc#aes128-cbc .
Security	SymmetricEncryptionAlgorithm_AES256-CBC	The AES encryption algorithm which is defined in http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf . Multiple blocks encrypted using the CBC mode described in http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf . The key size is 256 bits. The block size is 16 bytes. The URI is http://www.w3.org/2001/04/xmlenc#aes256-cbc .

Category	Title	Description
Security	SymmetricEncryptionAlgorithm_AES128-CTR	<p>The AES encryption algorithm which is defined in http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf.</p> <p>Multiple blocks encrypted using the CTR mode described in http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf.</p> <p>The counter block format is defined in https://tools.ietf.org/html/rfc3686.</p> <p>The key size is 128 bits. The block size is 16 bytes. The input nonce length is 4 bytes.</p> <p>The URI is http://opcfoundation.org/UA/security/aes128-ctr.</p>
Security	SymmetricEncryptionAlgorithm_AES256-CTR	<p>The AES encryption algorithm which is defined in http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf.</p> <p>The key size is 256 bits. The block size is 16 bytes.</p> <p>Multiple blocks encrypted using the CTR mode described in http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf.</p> <p>The counter block format is defined in https://tools.ietf.org/html/rfc3686.</p> <p>The key size is 128 bits. The block size is 16 bytes. The input nonce length is 4 bytes.</p> <p>The URI is http://opcfoundation.org/UA/security/aes256-ctr.</p>
Security	AsymmetricSignatureAlgorithm_None	This algorithm does not apply.
Security	AsymmetricSignatureAlgorithm_RSA- PKCS15-SHA1	<p>The RSA signature algorithm which is defined in https://tools.ietf.org/html/rfc3447.</p> <p>The RSASSA-PKCS1-v1_5 scheme is used.</p> <p>The hash algorithm is SHA1 and is described in https://tools.ietf.org/html/rfc3174.</p> <p>The URI is http://www.w3.org/2000/09/xmldsig#rsa-sha1.</p> <p>SHA1 was broken in 2017 so this algorithm should not be used.</p>
Security	AsymmetricSignatureAlgorithm_RSA-PKCS15-SHA2-256	<p>The RSA signature algorithm which is defined in https://tools.ietf.org/html/rfc3447.</p> <p>The RSASSA-PKCS1-v1_5 scheme is used.</p> <p>The hash algorithm is SHA2 with 256bits and is described in https://tools.ietf.org/html/rfc6234.</p> <p>The URI is http://www.w3.org/2001/04/xmldsig-more#rsa-sha256.</p>
Security	AsymmetricSignatureAlgorithm_RSA-PSS -SHA2-256	<p>The RSA signature algorithm which is defined in https://tools.ietf.org/html/rfc3447.</p> <p>The RSASSA-PSS scheme is used.</p> <p>The hash algorithm is SHA2 with 256bits and is described in https://tools.ietf.org/html/rfc6234.</p> <p>The mask generation algorithm also uses SHA2 with 256 bits.</p> <p>The salt length is 32 bytes.</p> <p>The URI is http://opcfoundation.org/UA/security/rsa-pss -sha2-256.</p>
Security	AsymmetricEncryptionAlgorithm_None	This algorithm does not apply.
Security	AsymmetricEncryptionAlgorithm_RSA-PKCS15	<p>The RSA encryption algorithm which is defined in https://tools.ietf.org/html/rfc3447.</p> <p>The RSAES-PKCS1-v1_5 scheme is used.</p> <p>The URI is http://www.w3.org/2001/04/xmlenc#rsa-1_5.</p> <p>The RSAES-PKCS1-v1_5 scheme has known weaknesses and is not recommended.</p>