IEC 62541-7

Edition 3.0 2020-06

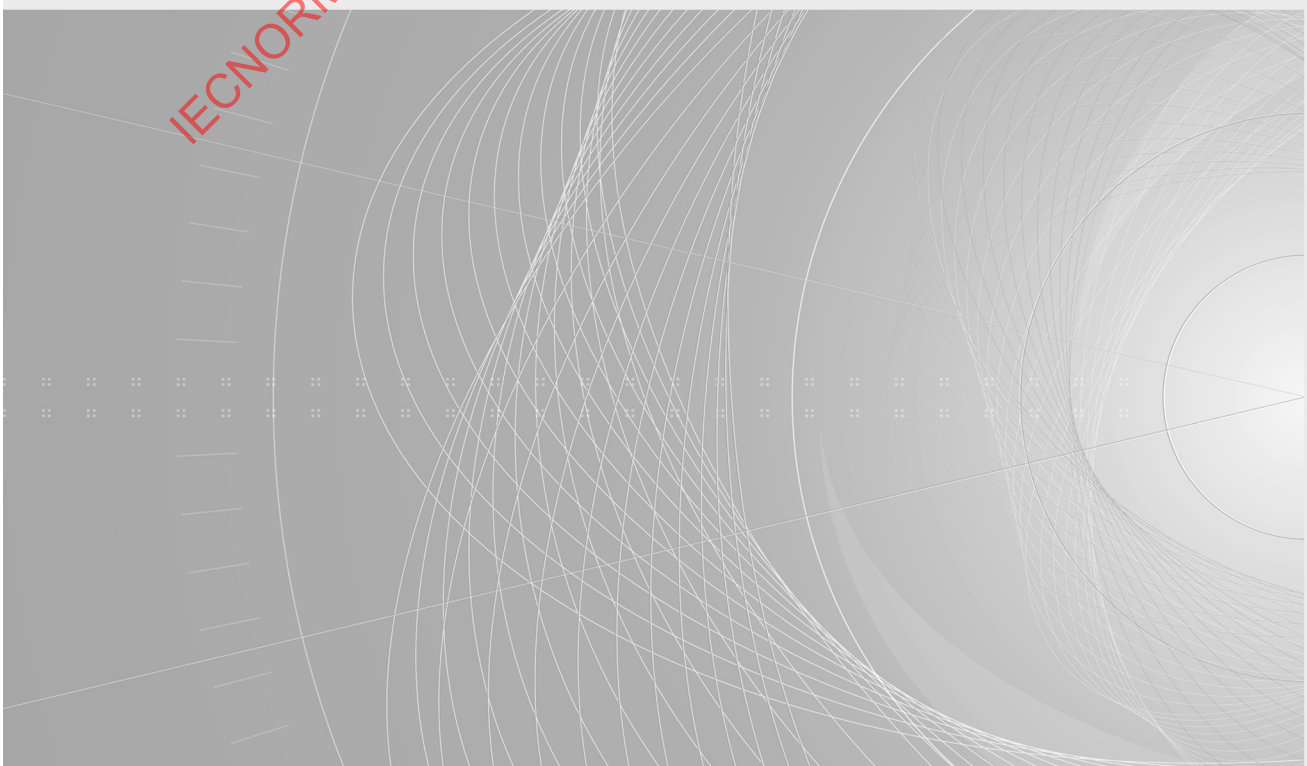
INTERNATIONAL STANDARD

NORME INTERNATIONALE

colour inside

OPC unified architecture – Part 7: Profiles

Architecture unifiée OPC – Partie 7: Profils



IEC 62541-7:2020-06(en-fr)

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

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

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

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

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 It also gives information on projects, replaced and withdrawn publications.

IEC Just Published - webstore.iec.ch/justpublished
Stay up to date on all new IEC publications. Just Published details all new publications released. Available online and once a month by email.

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

Electropedia - www.electropedia.org
The world's leading online dictionary on electrotechnology, containing more than 22 000 terminological entries in English and French, with equivalent terms in 16 additional languages. Also known as the International Electrotechnical Vocabulary (IEV) online.

IEC Glossary - std.iec.ch/glossary
67 000 electrotechnical terminology entries in English and French extracted from the Terms and Definitions clause of IEC publications issued since 2002. Some entries have been collected from earlier publications of IEC TC 37, 77, 86 and CISPR.

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

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

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

IEC Just Published - webstore.iec.ch/justpublished
Restez informé sur les nouvelles publications IEC. Just Published détaille les nouvelles publications parues. Disponible en ligne et une fois par mois par email.

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

Electropedia - www.electropedia.org
Le premier dictionnaire d'électrotechnologie en ligne au monde, avec plus de 22 000 articles terminologiques en anglais et en français, ainsi que les termes équivalents dans 16 langues additionnelles. Egalement appelé Vocabulaire Electrotechnique International (IEV) en ligne.

Glossaire IEC - std.iec.ch/glossary
67 000 entrées terminologiques électrotechniques, en anglais et en français, extraites des articles Termes et Définitions des publications IEC parues depuis 2002. Plus certaines entrées antérieures extraites des publications des CE 37, 77, 86 et CISPR de l'IEC.

IEC 62541-7

Edition 3.0 2020-06

INTERNATIONAL STANDARD

NORME INTERNATIONALE

colour inside

OPC unified architecture – Part 7: Profiles

Architecture unifiée OPC – Partie 7: Profils

INTERNATIONAL ELECTROTECHNICAL COMMISSION

COMMISSION ELECTROTECHNIQUE INTERNATIONALE

ICS 25.040.40; 35.100.05

ISBN 978-2-8322-8456-8

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

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

_____

OPC UNIFIED ARCHITECTURE –

Part 7: Profiles

FOREWORD

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

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

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

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

a) new functional Profiles:
   - profiles for global discovery and global certificate management;
   - profiles for global KeyCredential management and global access token management;
   - facet for durable subscriptions;
   - standard UA Client Profile;

- profiles for administration of user roles and permissions.

b) new transport Profiles:
   - HTTPS with JSON encoding;
   - secure WebSockets (WSS) with binary or JSON encoding;
   - reverse connectivity.

c) new security Profiles:
   - transportSecurity – TLS 1.2 with PFS (with perfect forward secrecy);
   - securityPolicy [A] – Aes128-Sha256-RsaOaep (replaces Base128Rsa15);
   - securityPolicy – Aes256-Sha256-RsaPss adds perfect forward secrecy for UA TCP);
   - user Token JWT (Jason Web Token).

d) deprecated Security Profiles (due to broken algorithms):
   - securityPolicy – Basic128Rsa15 (broken algorithm Sha1);
   - securityPolicy – Basic256 (broken algorithm Sha1);
   - transportSecurity – TLS 1.0 (broken algorithm RC4);
   - transportSecurity – TLS 1.1 (broken algorithm RC4).

e) deprecated Transport (missing support on most platforms):
   - SOAP/HTTP with WS-SecureConversation (all encodings).

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

FDIS	Report on voting
65E/707/FDIS	65E/725/RVD

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

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

Throughout this document and the other parts of the IEC 62541 series, certain document conventions are used:

Italics are used to denote a defined term or definition that appears in the "Terms and definition" clause in one of the parts of the IEC 62541 series.

Italics are also used to denote the name of a service input or output parameter or the name of a structure or element of a structure that are usually defined in tables.

The *italicized terms and names* are, with a few exceptions, written in camel-case (the practice of writing compound words or phrases in which the elements are joined without spaces, with each element's initial letter capitalized within the compound). For example the defined term is *AddressSpace* instead of Address Space. This makes it easier to understand that there is a single definition for *AddressSpace*, not separate definitions for Address and Space.

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

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

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

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

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.

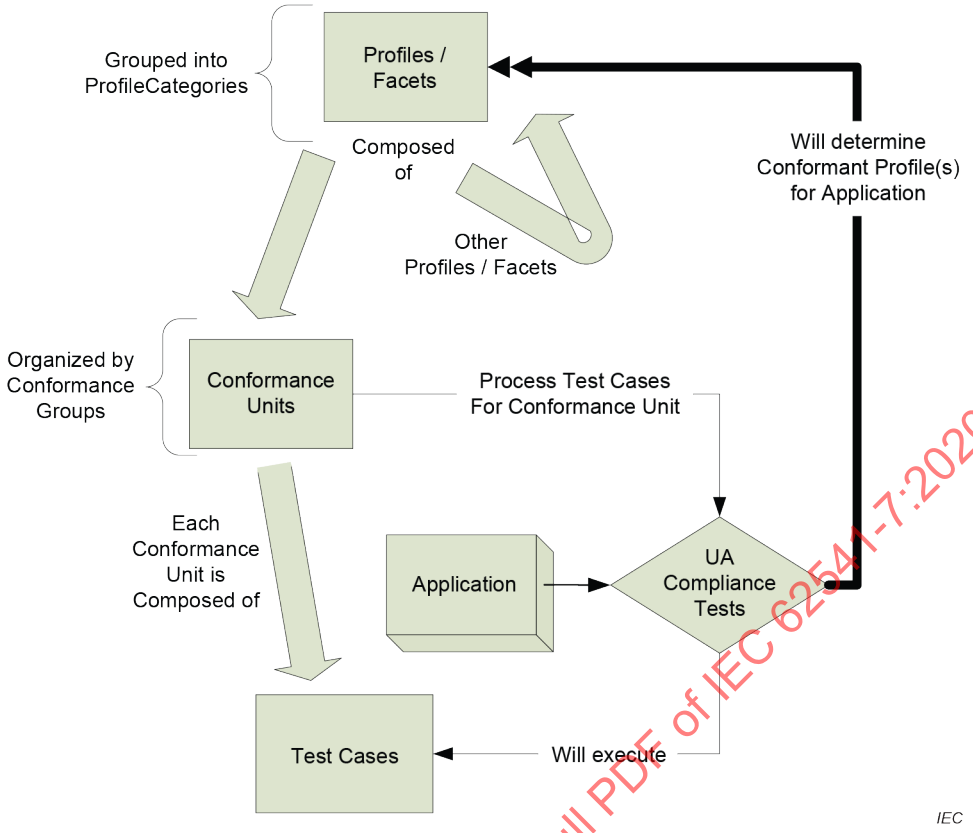


Figure 1 – Profile – ConformanceUnit – TestCases

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

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	*Discovery* Get Endpoints	Support the GetEndpoints *Service* to obtain all Endpoints of the *Server*. This includes filtering based on *Profiles*.
Server	*Discovery* Get Endpoints SessionLess	Support at least one endpoint for issueing SessionLess Services. Support obtaining such endpoints by accepting the Transport URI as a filter to the GetEndpoints Service with the query string "SL" appended to the Transport URI. E.g. "http://opcfoundation.org/UA-Profile/Transport/https-uajson?SL"
Server	*Discovery* Find *Servers* Self	Support the FindServers *Service* only for itself.
Server	*Discovery* Register	Call the RegisterServer *Service* to register itself (OPC UA *Server*) with an external *Discovery Service* via a secure channel with a SecurityMode other than NONE.
Server	*Discovery* Register2	Call the RegisterServer2 *Service* to register with an external *Discovery Service* 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	*Discovery Server* Announcement using mDNS	Provide mDNS functionality to announce a *Server* with its capabilities. The capability identifiers and the use of mDNS records for the purpose of OPC UA *Discovery* is specified in IEC 62541-12. Note that this functionality is only required for *Servers* that do not register with an LDS. The capability identifiers and their use in mDNS records are specified in IEC 62541-12.
Server	*Discovery* Configuration	Allow configuration of the *Discovery Server* URL where the *Server* will register itself. Allow complete disabling of registration with a *Discovery Server*.
Client	*Discovery Client* Find *Servers* Basic	Uses the FindServers *Service* to obtain all *Servers* installed on a given platform.
Client	*Discovery Client* Find *Servers* with URI	Use FindServers *Service* to obtain URLs for specific *Server* URIs.
Client	*Discovery Client* Find *Servers* Dynamic	Detect new *Servers* after an initial FindServers *Service* call.
Client	*Discovery Client* Find *Servers* on Network	Support one of the options to locate *Servers* on the network.
Client	*Discovery Client* Find *Servers* on Network using LDS-ME	Use FindServersOnNetwork *Service* to obtain URLs for specific *Server* URIs. Note that this *Service* is available via the Local *Discovery Server* with multicast extension (LDS-ME).
Client	*Discovery Client* Find *Servers* on Network using mDNS	Use mDNS based *Service Discovery* to locate *Servers* on the same multicast network. The contents of mDNS records for OPC UA *Discovery* are described in IEC 62541-12. Note that this functionality is only required for *Clients* when there is no Local *Discovery Server* with multicast extension (LDS-ME). The capability identifiers and their use in mDNS records are specified in IEC 62541-12.
Client	*Discovery Client* Find *Servers* in GDS	Use the QueryServers *Method* on the GDS Directory *Object* to locate *Servers* that meet filter criteria specified in the request. This *Method* is specified in IEC 62541-12.
Client	*Discovery Client* Find Applications in GDS	Use the QueryApplications *Method* on the GDS Directory *Object* to locate Applications that meet filter criteria specified in the request. This *Method* is specified in IEC 62541-12.

Category	Title	Description
Client	*Discovery Client* Get Endpoints Basic	Uses the GetEndpoints *Service* to obtain all Endpoints for a given *Server* URI.
Client	*Discovery Client* Get Endpoints SessionLess	Uses the GetEndpoints *Service* with a filter to obtain Endpoints that can be used for SessionLess *Service* invocation. The filter is the Transport URI extended with the query string "SL". E.g. http://opcfoundation.org/UA-*Profile*/Transport/https-uajson?SL.
Client	*Discovery Client* Get Endpoints Dynamic	Detect changes to the Endpoints after an initial GetEndpoints *Service* call.
Client	*Discovery Client* Configure Endpoint	Allow specification of an Endpoint without going through the *Discovery Service* 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
Server	*Session* General *Service* Behaviour	Implement basic *Service* behaviour. This includes in particular: – checking the authentication token – returning the requestHandle in responses – returning available diagnostic information as requested with the 'returnDiagnostics' parameter – respecting a timeoutHint
Server	*Session* Base	Support the *Session Service* Set (CreateSession, ActivateSession, CloseSession) except the use of ActivateSession to change the *Session* 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 *Certificate* and Nonce are optional. 2) The signatures are null/empty. The details of this are described in IEC 62541-4.
Server	*Session* Change User	Support the use of ActivateSession to change the *Session* user.
Server	*Session* Cancel	Support the Cancel *Service* to cancel outstanding requests.
Server	*Session* Minimum 1	Support minimum 1 *Session* (total).
Server	*Session* Minimum 2 Parallel	Support minimum 2 parallel Sessions (total for all *Clients*).
Server	*Session* Minimum 50 Parallel	Support minimum 50 parallel Sessions (total for all *Clients*).
Server	Session Sessionless Invocation	Defines the support of the SessionlessInvoke Service defined in IEC 62541-4 to process any of the Services (like Read/Write, Browse, or Call) that are designated for Sessionless Invocation.
Client	*Session Client* General *Service* Behaviour	Implement basic *Service* behaviour. This includes in particular: – including the proper authentication token of the *Session* – creating a requestHandle if needed – requesting diagnostic information with the 'returnDiagnostics' parameter – evaluate the serviceResult and operational results
Client	*Session Client* Base	Use the *Session Service* Set (CreateSession, ActivateSession, and CloseSession) except the use of ActivateSession to change the *Session* 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 *Certificate* and Nonce are optional. 2) The signatures are null/empty.

Category	Title	Description
Client	*Session Client* Multiple Connections	Support unlimited connections (client side) with multiple *Servers*. Any limit on numbers of connections is from server side. May have a memory-based limit, but not a software constraint limit.
Client	*Session Client* Renew NodeIds	This *ConformanceUnit* applies to *Clients* that allow persisting NodeIds. Verify that the Namespace Table has not changed for NodeIds that the *Client* has persisted and is going to re-use beyond a *Session* lifetime. If changes occurred the *Client* has to recalculate the Namespace Indices of the respective NodeIds.
Client	*Session Client* Impersonate	Uses ActivateSession to change the Session user (impersonation).
Client	*Session Client* KeepAlive	Make periodic requests to keep the *Session* alive.
Client	*Session Client* Detect Shutdown	Read or monitor the ServerStatus/State Variable to recognize a potential shutdown of the Server and clean up resources.
Client	*Session Client* Cancel	Use the Cancel *Service* to cancel outstanding requests.
Client	*Session Client* Auto Reconnect	Automatic *Client* reconnect including: – ActivateSession with new SecureChannel if SecureChannel is no longer valid but *Session* is still valid – Creation of a new *Session* only if *Session* is no longer valid
Client	*Session Client* Single *Session*	The *Client* shall interoperate with *Servers* that only support one *Session*.
Client	*Session Client* SessionLess *Service* Calls	Defines the use of the SessionlessInvoke *Service* defined in IEC 62541-4 to request one of the *Services* (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 *Service* to add one or more *Nodes* into the OPC UA *AddressSpace*.
Server	*Node* Management Delete *Node*	Support the DeleteNodes *Service* to delete one or more *Nodes* from the OPC UA *AddressSpace*.
Server	*Node* Management Add Ref	Support the AddReferences *Service* to add one or more *References* to one or more *Nodes* in the OPC UA *AddressSpace*.
Server	*Node* Management Delete Ref	Support the DeleteReferences *Service* to delete one or more *References* of a *Node* in the OPC UA *AddressSpace*.
Client	*Node* Management *Client*	Uses *Node* Management *Services* to add or delete *Nodes* and to add or delete *References* in *Server*'s OPC UA *AddressSpace*.

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 *Service* Set (Browse, BrowseNext).
Server	View TranslateBrowsePath	Support TranslateBrowsePathsToNodeIds *Service*.
Server	View RegisterNodes	Support the RegisterNodes and UnregisterNodes *Services* as a way to optimize access to repeatedly used *Nodes* in the *Server*'s OPC UA *AddressSpace*.
Server	View Minimum Continuation Point 01	Support minimum 1 continuation point per *Session*.
Server	View Minimum Continuation Point 05	Support minimum 5 continuation points per *Session*. This number has to be supported for at least half of the minimum required sessions.
Client	View *Client* Basic Browse	Uses Browse and BrowseNext *Services* to navigate through the *Server*'s OPC UA *AddressSpace*. Make use of the referenceTypeId and the nodeClassMask to specify the needed *References*.
Client	View *Client* Remote *Nodes* Browse	The *Client* can browse to nodes that have an extended NodeID that reference a *Server* different than the originating *Server*. This includes automatic connection to the remote *Server*. It is acceptable that the *Server* configuration information be pre-configured on the *Client* 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 *Server*.
Client	View *Client* TranslateBrowsePath	Uses the TranslateBrowsePathsToNodeIds *Service* to identify the NodeIds for *Nodes* where a starting *Node* and a BrowsePath is known. Makes use of bulk operations rather than multiple calls whenever possible.
Client	View *Client* Remote *Nodes* Translate Browse	The *Client* can translate browse paths that include nodes with extended NodeID that reference a *Server* different than the originating *Server* and return them as part of the TranslateBrowsePathsToNodeIds *Service*. It is acceptable that the *Server* configuration information be pre-configured on the *Client*.
Client	View *Client* RegisterNodes	Uses the RegisterNodes *Service* to optimize access for *Nodes* that are used repeatedly. Use UnregisterNodes when *Nodes* 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	*Attribute* Read	Supports the Read *Service* to read one or more *Attributes* of one or more *Nodes*. This includes support of the IndexRange parameter to read a single element or a range of elements when the *Attribute* value is an array.
Server	*Attribute* Read Complex	Supports reading and encoding Values with structured DataTypes.
Server	*Attribute* Write Values	Supports writing to values to one or more *Attributes* of one or more *Nodes*.
Server	*Attribute* Write Complex	Supports writing and decoding Values with structured DataTypes.
Server	*Attribute* Write StatusCode & Timestamp	Supports writing of StatusCode and Timestamps along with the Value.
Server	*Attribute* Write Index	Supports the IndexRange to write a single element or a range of elements when the *Attribute* value is an array and partial updates are allowed for this array.
Server	*Attribute* Alternate Encoding	Supports alternate Data Encoding when reading value *Attributes*. By default, every *Server* has to support the Data Encoding of the currently used Stack *Profile* (i.e. binary with UA Binary Encoding and XML with XML Encoding). This *ConformanceUnit* – when supported – specifies that the other Data Encoding is supported in addition.
Server	*Attribute* Historical Read	Supports the HistoryRead *Service*. The details of what aspects of this service are used are listed in additional *ConformanceUnits*, but at least one of ReadRaw, ReadProcessed, ReadModified, ReadAtTime or ReadEvents shall be supported.
Server	*Attribute* Historical Update	Supports the HistoryUpdate service. The details of the supported features of this service are described by additional *ConformanceUnits*, but at least one of the following shall be supported: InsertData, InsertEvents, ReplaceData, ReplaceEvents, UpdateData, UpdateEvents, DeleteData, DeleteEvents or DeleteAtTime.
Client	*Attribute Client* Read Base	Use the Read *Service* to read one or more *Attributes* of one or more *Nodes*. This includes use of an IndexRange to select a single element or a range of elements when the *Attribute* value is an array. *Clients* shall use bulk operations whenever possible to reduce the number of *Service* invocations.
Client	*Attribute Client* Remote *Nodes Attribute* Access	The *Client* can retrieve attributes of nodes that have an extended NodeID that reference a *Server* different than the originating *Server*. This requires a connection to the remote *Server* for access (not necessarily displayed as a connection). It is acceptable that the *Server* configuration information be pre-configured on the *Client*.
Client	*Attribute Client* Read with proper Encoding	This *ConformanceUnit* refers to the ability of a *Client* to discover the available encodings and choose a specific one when calling the Read *Service*.
Client	*Attribute Client* Read Complex	Read and decode Values with structured DataTypes.
Client	*Attribute Client* Write Base	Use the Write *Service* to write values to one or more *Attributes* of one or more *Nodes*. This includes use of an IndexRange to select a single element or a range of elements when the *Attribute* value is an array. *Clients* shall use bulk operations whenever possible to reduce the number of *Service* invocations.
Client	*Attribute Client* Write Complex	Write and Encode Values with structured DataTypes.
Client	*Attribute Client* Write Quality & Timestamp	Use the Write *Service* to also write StatusCode and/or Timestamps along with a Value.
Client	*Attribute Client* Historical Read	The *Client* makes use of the HistoryRead service. The details of which aspect of this service are used are provided by additional *ConformanceUnits*, but at least one or more of the following is used ReadRaw, ReadAtTime, ReadProcessed, ReadModified or ReadEvents.

Category	Title	Description
Client	*Attribute Client* Historical Updates	The *Client* makes use of the HistoryUpdate service. The details of this usage are provided by additional *ConformanceUnits*, 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
Server	*Method* Call	Support the Call *Service* to call (invoke) a *Method* which includes support for *Method* Parameters.
Client	*Method Client* Call	Use the Call *Service* to call one or several Methods.

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
Server	Monitor Basic	Support the following *MonitoredItem Services*: CreateMonitoredItems, ModifyMonitoredItems, DeleteMonitoredItems and SetMonitoringMode.
Server	Monitor Value Change	Support creation of *MonitoredItems* for *Attribute* value changes. This includes support of the IndexRange to select a single element or a range of elements when the *Attribute* value is an array.
Server	Monitor Complex Value	Supports monitoring and encoding Values with structured DataTypes.
Server	Monitored Items Deadband Filter	Supports an absolute Deadband filter as a DataChangeFilter for numeric data types.
Server	Monitor Aggregate Filter	Support for Aggregate filters for *MonitoredItems*. The result of this *ConformanceUnit* includes a list of Aggregates that are supported as part of the *Profile Certificate*.
Server	Monitor Alternate Encoding	Support alternate encoding when monitoring value *Attributes*. By default, every *Server* has to support the encoding of the currently used Stack *Profile* (i.e. binary with UA Binary Encoding and XML with XML Encoding). This *ConformanceUnit* – when supported – specifies that the other encoding is supported in addition.
Server	Monitor Items 2	Support at least 2 *MonitoredItems* per *Subscription* where the size of each *MonitoredItem* is at least equal to size of Double.
Server	Monitor Items 10	Support at least 10 *MonitoredItems* per *Subscription* where the size of each *MonitoredItem* is at least equal to size of Double.
Server	Monitor Items 100	Support at least 100 *MonitoredItems* per *Subscription*. 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 *MonitoredItems* per *Subscription*. 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 *ConformanceUnit* does not require queuing when multiple value changes occur during a "publish period". I.e. the latest change will be sent in the *Notification*.
Server	Monitor MinQueueSize_02	Support at least 2 queue entries for *MonitoredItems*. *Servers* often will adapt the queue size to the number of currently *MonitoredItems*. However, it is expected that *Servers* support this minimum queue size for at least one third of the supported *MonitoredItems*.
Server	Monitor MinQueueSize_05	Support at least 5 queue entries for *MonitoredItems*. *Servers* often will adapt the queue size to the number of currently *MonitoredItems*. However, it is expected that *Servers* support this minimum queue size for at least one third of the supported *MonitoredItems*.
Server	Monitor QueueSize_ServerMax	This *ConformanceUnit* is for events. When the *Client* requests queuesize=MAXUInt32 the *Server* is to return the maximum queue size that it can support for event notifications as the revisedQueueSize.
Server	Monitor Triggering	Support the SetTriggering *Service* to create and/or delete triggering links for a triggering item.
Server	Monitor Events	Support creation of *MonitoredItems* for an "*EventNotifier Attribute*" for the purpose of *Event Notification*. 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, LessThatorEqual, Like, Not, Between, InList, And, Or, Cast, BitwiseAnd, BitwiseOr.
Server	Monitor Complex *Event* Filter	Support for the 'TypeOf' complex *Event* filter operator.

Category	Title	Description
Client	Monitor *Client* Value Change	Use the *MonitoredItem Service* Set to register items for changes in *Attribute* value. Use CreateMonitoredItems to register the *Node*/*Attribute* tuple. Set proper sampling interval, Deadband filter and queuing mode. Use disabling / enabling instead of deleting and re-creating a *MonitoredItem*. 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 *Attribute* value is an array.
Client	Monitor *Client* Aggregate Filter	Uses Aggregate filters for Subscriptions.
Client	Monitor *Client* Events	Use the *MonitoredItem Service* Set to create *MonitoredItems* for *Event* notifications.
Client	Monitor *Client Event* Filter	Use the *Event* filter when calling CreateMonitoredItems to filter the desired Events and to select the columns to be provided for each *Event Notification*.
Client	Monitor *Client* Complex *Event* Filter	Use of the 'TypeOf' complex *Event* filter operator.
Client	Monitor *Client* Modify	Use ModifyMonitoredItems *Service* to change the configuration setting. Use SetMonitoringMode *Service* to disable/enable sampling and/or publishing.
Client	Monitor *Client* Trigger	Use the Triggering Model if certain items are to be reported only if some other item triggers. Use proper monitoring mode for these items. Use SetTriggering *Service* 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
Server	*Subscription* Basic	Support the following *Subscription Services*: CreateSubscription, ModifySubscription, DeleteSubscriptions, Publish, Republish and SetPublishingMode.
Server	*Subscription* Minimum 1	Support at least 1 *Subscription* per *Session*. This number has to be supported for all of the minimum required sessions.
Server	*Subscription* Minimum 02	Support at least 2 Subscriptions per *Session*. This number has to be supported for at least half of the minimum required sessions.
Server	*Subscription* Minimum 05	Support at least 5 Subscriptions per *Session*. This number has to be supported for at least half of the minimum required sessions.
Server	*Subscription* Publish Min 02	Support at least 2 Publish *Service* requests per *Session*. This number has to be supported for all of the minimum required sessions. Support of a NotificationMessage retransmission queue is not required; if not available, the Republish *Service* returns Bad_MessageNotAvailable.
Server	*Subscription* Publish Min 05	Support at least 5 Publish *Service* requests per *Session*. This number has to be supported for at least half of the minimum required sessions. Support, as a minimum, the number of Publish requests per session as the size of the NotificationMessage retransmission queue for Republish.
Server	*Subscription* Publish Min 10	Support at least 10 Publish *Service* requests per *Session*. This number has to be supported for at least half of the minimum required sessions. Support, as a minimum, the number of Publish requests per session as the size of the NotificationMessage retransmission queue for Republish.
Server	*Subscription* Publish Discard Policy	Respect the specified policy for discarding Publish *Service* requests. If the maximum number of Publish *Service* requests has been queued and a new Publish *Service* request arrives, the "oldest" Publish request has to be discarded by returning the proper error.
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.

Category	Title	Description
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 *Server* 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 *Server* 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	The *Server* supports a Kerberos *Server* 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. 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 IssuedToken Kerberos Windows	The *Server* supports the Windows implementation of Kerberos Tokens. This *ConformanceUnit* only applies if the "Security User IssuedToken Kerberos" is supported. 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 JWT IssuedToken	The *Server* 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. 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 Anonymous	The *Server* 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 *Client*	A *Client* uses a User Name/Password combination. 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 *Client*	A *Client* 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 *Client*	A *Client* uses a Kerberos *Server* token. The use of this token is defined by the Kerberos documentation. 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.

Category	Title	Description
Security	Security User IssuedToken Kerberos Windows *Client*	A *Client* uses the Windows implementation of Kerberos tokens. This *ConformanceUnit* only applies if the "Security User IssuedToken Kerberos *Client*" is supported. 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 JWT IssuedToken *Client*	A *Client* uses a JSON Web Token (JWT) for user identity. IEC 62541-6 describes OAuth2 and JWTs in more detail. 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 Invalid user token	*Servers* 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 ActivateSession *Service* in IEC 62541-4.
Security	Security User JWT Token Policy	The Server supports one or more Endpoints with a UserTokenPolicy that includes a JWT IssuerEndpointUrl as defined in IEC 62541-6. For JWT the issuerEndpointUrl is a JSON object that includes all parameters that define the AuthorizationService. 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.
Security	Security User JWT Token Policy *Client*	The Client understands and uses the Authorization Service definition inside the JWT UserTokenPolicy returned with GetEndpoints. 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.
Security	OAuth2 Authority *Profile*	This unit indicates support of OAuth2 over HTTPS to request access tokens. The URI for the interactions with this authority is "http://opcfoundation.org/UA/Authorization#OAuth2"
Security	OPC UA Authority *Profile*	This unit indicates support of the OPC UA *Methods* defined in IEC 62541-12 to request access tokens. The URI for the interactions with this authority is "http://opcfoundation.org/UA/Authorization#OPCUA"
Security	Azure Identity Provider Authority *Profile*	This unit indicates support of the Azure identity provider to request access tokens. The URI for the interactions with this authority is "http://opcfoundation.org/UA/Authorization#Azure"
Security	Security *Certificate* Validation	A certificate will be validated as specified in IEC 62541-4. This includes, among others, structure and signature examination. Allowing for some validation errors to be suppressed by administration directive.
Security	Security Default ApplicationInstance Certificate	An application, when installed, has a default ApplicationInstanceCertificate that is valid. The default ApplicationInstanceCertificate shall either be created as part of the installation or installation instructions explicitly describe the process to create and apply a default ApplicationInstanceCertificate to the application.
Security	Security – No Application Authentication	The *Server* supports being able to be configured for no application authentication, just User authentication and normal encryption/signing: – Configure *Server* to accept all certificates – *Certificates* are just used for message security (signing and encryption) – Users level is used for authentication

Category	Title	Description
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 *Certificates* or a Nonce.
Security	Security None CreateSession ActivateSession 1.0	The *Client* can connect to *Servers* that require a certificate being passed on Session establishment. The *Client* in this case will first try without a *Certificate* and if this fails present a *Certificate*.
Security	Security TLS General	This *ConformanceUnit* indicates that at least one of the transport security *Profiles* for TLS is supported by this application. It is used in TLS transport *Profiles*, 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). *Clients* and *Servers* 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 *Discovery Server* to synchronize the time on the application and that this time synchronization occurs periodically. Use of this TimeSyncing can be configured.
Security	Security Administration	Allow configuration of the following Security related items (when they apply). * 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 *Certificate* Administration	Allow a site administrator to be able to assign a site specific ApplicationInstanceCertificate and if desired to configure a site-specific *Certificate* Authority (CA).
Security	Security Role *Server* Base	Support the User Authorization Information Model defined in UA Parts 3 and 5 – like Roles – and the RolePermissions and UserRolePermissions *Attributes*.
Security	Security Role Well Known	Support the well-known Roles "ConfigureAdmin" and "SecurityAdmin" with suggested permissions defined in IEC 62541-3.
Security	Security Role *Server* IdentityManagement	Allow authorized users to add and/or remove Identities from Roles with the appropriate *Methods*.

Category	Title	Description
Security	Security Role *Server* Management	Allow authorized users to create new Roles and/or remove Roles with the appropriate Methods.
Security	Security Role *Server* Restrict Applications	Support adding applications to a Role with the appropriate *Methods* so that only these applications can use this Role.
Security	Security Role *Server* Restrict Endpoints	Support adding Endpoints to a Role with the appropriate *Methods*. With this restriction a Role is only applied when a *Client* connects via one of these Endpoints.
Security	Security Role *Server* DefaultRolePermissions	Allow authorized users to set the DefaultRolePermissions *Property* for certain NameSpaces. DefaultRolePermissions are applied if no RolePermissions are associated with a *Node*.
Security	Security Role *Server* RolePermissions	Allow authorized users to set the RolePermissions *Attribute* on *Nodes*.
Security	Security Role *Server* Authorization	Restrict access based on the configured Roles and permissions.
Security	Security Role *Client* Base	Understand and use the User Authorization Information Model defined in IEC 62541-5 and the RolePermissions *Attribute*.
Security	Security Role *Client* Management	Support creating new Roles and adding Identities as well as remove Roles or Identities using the appropriate *Methods*.
Security	Security Role *Client* Restrict Applications	Use the appropriate *Methods* to add applications to a Role so that only these applications can use this Role.
Security	Security Role *Client* Restrict Endpoints	Use the appropriate *Methods* to add Endpoints to a Role. With this restriction a Role is only applied when a *Client* connects via one of these Endpoints.
Security	Security Role *Client* DefaultRolePermissions	Ability to set the DefaultRolePermissions *Property* for certain NameSpaces. DefaultRolePermissions are applied if no RolePermissions are associated with a *Node*.
Security	Security Role *Client* RolePermissions	Support setting the RolePermissions *Attribute* on *Nodes*.
Security	Pull Model for Global *Certificate* and TrustList Management	Use the *Certificate* Management *Services* of IEC 62541-12 for the Pull model to manage Application Instance *Certificates* and Trust Lists including Revocation Lists.
Security	Push Model for Global *Certificate* and TrustList Management	Support the *Certificate* Management *Services* of IEC 62541-12 for the Push model to manage Application Instance *Certificates* and Trust Lists including Revocation Lists.
Security	Pull Model for KeyCredential *Service*	Use the *Methods* on an instance of the KeyCredentialServiceType (Pull model) to obtain KeyCredentials as specified in IEC 62541-12.
Security	Push Model for KeyCredential *Service*	Support the KeyCredential *Services* Push model of IEC 62541-12 to obtain KeyCredentials. This includes support of one or more instances of the KeyCredentialConfigurationType and the *Methods* 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/xmldsig#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

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

Category	Title	Description
Security	AsymmetricSignatureAlgorithm_RSA-PSS - SHA2-256	The RSA signature algorithm which is defined in https://tools.ietf.org/html/rfc3447. The RSASSA-PSS scheme is used. The hash algorithm is SHA2 with 256bits and is described in https://tools.ietf.org/html/rfc6234. The mask generation algorithm also uses SHA2 with 256 bits. The salt length is 32 bytes. The URI is http://opcfoundation.org/UA/security/rsa-pss -sha2-256.
Security	AsymmetricEncryptionAlgorithm_None	This algorithm does not apply.
Security	AsymmetricEncryptionAlgorithm_RSA-PKCS15	The RSA encryption algorithm which is defined in https://tools.ietf.org/html/rfc3447. The RSAES-PKCS1-v1_5 scheme is used. The URI is http://www.w3.org/2001/04/xmlenc#rsa-1_5. The RSAES-PKCS1-v1_5 scheme has known weaknesses and is not recommended.
Security	AsymmetricEncryptionAlgorithm_RSA-OAEP-SHA1	The RSA encryption algorithm which is defined in https://tools.ietf.org/html/rfc3447. The RSAES-OAEP scheme is used. The hash algorithm is SHA1 and is described in https://tools.ietf.org/html/rfc6234. The mask generation algorithm also uses SHA1. The URI is http://www.w3.org/2001/04/xmlenc#rsa-oaep. No known exploits exist when using SHA1 with RSAES-OAEP, however, SHA1 was broken in 2017 so use of this algorithm is not recommended.
Security	AsymmetricEncryptionAlgorithm_RSA-OAEP-SHA2-256	The RSA encryption algorithm which is defined in https://tools.ietf.org/html/rfc3447. The RSAES-OAEP scheme is used. The hash algorithm is SHA2 with 256 bits and is described in https://tools.ietf.org/html/rfc6234. The mask generation algorithm also uses SHA2 with 256 bits. The URI is http://opcfoundation.org/UA/security/rsa-oaep-sha2-256.
Security	KeyDerivationAlgorithm_None	This algorithm does not apply.
Security	KeyDerivationAlgorithm_P-SHA1	The P_SHA-1 pseudo-random function defined in https://tools.ietf.org/html/rfc4346. The URI is http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/dk/p_sha1. No known exploits exist when using SHA1 with P-SHA-1, however, SHA1 was broken in 2017 so use of this algorithm is not recommended.
Security	KeyDerivationAlgorithm_P-SHA2-256	The P_SHA256 pseudo-random function defined in https://tools.ietf.org/html/rfc5246. The URI is http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/dk/p_sha256.
Security	CertificateSignatureAlgorithm_None	This algorithm does not apply.

Category	Title	Description
Security	CertificateSignatureAlgorithm_RSA-PKCS15-SHA2-256	The RSA signature algorithm which is defined in https://tools.ietf.org/html/rfc3447. The RSASSA-PKCS1-v1_5 scheme is used. The hash algorithm is SHA2 with 256bits and is described in https://tools.ietf.org/html/rfc6234. The SHA2 algorithm with 384 or 512 bits may be used instead of SHA2 with 256 bits. The URI is http://www.w3.org/2001/04/xmldsig-more#rsa-sha256.
Security	CertificateSignatureAlgorithm_RSA-PKCS15-SHA1	The RSA signature algorithm which is defined in https://tools.ietf.org/html/rfc3447. The RSASSA-PKCS1-v1_5 scheme is used. The hash algorithm is SHA1 and is described in https://tools.ietf.org/html/rfc3174. The URI is http://www.w3.org/2000/09/xmldsig#rsa-sha1. SHA1 was broken in 2017 so this algorithm should not be used. The SHA2 algorithm with 244, 256, 384 or 512 bits may be used instead of SHA1. The SHA2 algorithm is described in https://tools.ietf.org/html/rfc6234.
Security	SecurtyPolicy_None_Limits	DerivedSignatureKeyLength: 0
Security	Aes128-Sha256-RsaOaep_Limits	-> DerivedSignatureKeyLength: 256 bits -> MinAsymmetricKeyLength: 2048 bits -> MaxAsymmetricKeyLength: 4096 bits -> SecureChannelNonceLength: 32 bytes
Security	Basic256Sha256_Limits	-> DerivedSignatureKeyLength: 256 bits -> MinAsymmetricKeyLength: 2048 bits -> MaxAsymmetricKeyLength: 4096 bits -> SecureChannelNonceLength: 32 bytes
Security	Aes256-Sha256-RsaPss_Limits	-> DerivedSignatureKeyLength: 256 bits -> MinAsymmetricKeyLength: 2048 bits -> MaxAsymmetricKeyLength: 4096 bits -> SecureChannelNonceLength: 32 bytes
Security	Basic128Rsa15_Limits	-> DerivedSignatureKeyLength: 128 bits -> MinAsymmetricKeyLength: 1024 bits -> MaxAsymmetricKeyLength: 2048 bits -> SecureChannelNonceLength: 16 bytes
Security	Basic256_Limits	-> DerivedSignatureKeyLength: 192 bits -> MinAsymmetricKeyLength: 1024 bits -> MaxAsymmetricKeyLength: 2048 bits -> SecureChannelNonceLength: 32 bytes

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

Table 12 – Protocol and Encoding

Category	Title	Description
Server	Protocol Reverse Connect *Server*	Support reverse connectivity by sending a ReverseHello message to a Client. The reverse connect procedure can be applied to several transports as specified in IEC 62541-6 and shall be supported for all of these that are available in a Server.
Server	Protocol Configuration	Allow administration of the Endpoints and the port number used by the Endpoints.
Client	Protocol Reverse Connect *Client*	Support reverse connectivity by accepting Reverse Hello messages from Servers and establish a Secure Channel if the URI of the Server is accepted by Client or Client user. The reverse connect procedure can be applied to several transports as specified in IEC 62541-6 and shall be supported for all of these transports that are supported by the Client.
Transport	Protocol UA TCP	Support the UA TCP transport protocol as defined in IEC 62541-6.
Transport	Protocol HTTPS	Support the HTTPS protocol as defined in IEC 62541-6.
Transport	Protocol Web Sockets	Support the WebSocket protocol (WSS) with at least one of the sub-protocols defined in IEC 62541-6.
Transport	UA Secure Conversation	Support UA Secure Conversation specified in IEC 62541-6.
Transport	UA Binary Encoding	Support UA Binary Encoding. Values of these data types are encoded in compact binary formats, contiguously and without tagging. I.e. the receiver is assumed to understand the structure in its decoding.
Transport	UA SOAP-XML Encoding	Support Soap V1.2 based Xml Encoding as defined in IEC 62541-6. The XML elements include all information necessary to convert it back into OPC UA structures of any language.
Transport	JSON Reversible Encoding	Support reversible JSON Encoding as defined in IEC 62541-6. The JSON object includes all information necessary to convert it back into OPC UA structures of any language.

5.4 Information Model and AddressSpace related features

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

Table 13 – Base Information

Category	Title	Description
Server	Base Info Core Structure	The *Server* supports the *Server Object*, ServerCapabilities and supports the OPC UA *AddressSpace* structure.
Server	Base Info *Server* Capabilities	The *Server* supports publishing of the *Server* limitation in the ServerCapabilities, including MaxArrayLength, MaxStringLength, MaxNodePerRead, MaxNodesPerWrite, MaxNodesPerSubscription and MaxNodesPerBrowse.
Server	Base Info Progress Events	The *Server* exposes if generation of Progress events for long running service calls such as HistoryRead or Query is supported. If it is listed as supported in ServerCapabilities, then the actual events are verified.
Server	Base Info Diagnostics	The *Server* supports the collection of diagnostic information. The EnabledFlag in the ServerDiagnostics *Object* can be set TRUE and in that case all static and dynamic *Objects* and *Variables* for diagnostic data as defined in IEC 62541-5 are supported.
Server	Base Info System Status	The *Server* supports generating SystemStatusChangeEventType indicating shutdown of the *Server* (SourceNode=*Server*).
Server	Base Info Estimated Return Time	*Server* supports the EstimatedReturnTime *Property*. It indicates the time at which the *Server* is expected to have a ServerStatus.State of RUNNING_0. *Clients* can use this information to govern the reconnect logic.
Server	Base Info System Status Underlying System	The *Server* supports generating SystemStatusChangeEventType indicating changes to an Underlying System (SourceNode = *Server*). This event can also be used to indicate that the OPC UA *Server* has underlying systems.
Server	Base Info Device Failure	The *Server* supports generating DeviceFailureEventType indicating changes to individual devices in an underlying system.
Server	Base Info GetMonitoredItems *Method*	The *Server* supports obtaining subscription information via GetMonitoredItems *Method* on the *Server* object.
Server	Base Info ResendData *Method*	Support the standard *Method* ResendData (defined in IEC 62541-5) to get the latest value of the monitored items of a *Subscription*.
Server	Base Info Type System	The *Server* exposes a Type System with DataTypes, ReferenceTypes, *ObjectTypes* and VariableTypes including all of the OPC UA (namespace 0) types that are used by the *Server*, as defined in IEC 62541-5. Items that are defined in Namespace 0 but are defined in other specification parts are tested as part of the other information models.
Server	Base Info Custom Type System	The *Server* supports custom types (i.e. types that are derived from well-known *ObjectTypes*, VariableTypes, ReferenceTypes or DataTypes). Supporting this conformance unit requires that the custom types with their full inheritance tree are exposed in the *AddressSpace*.
Server	Base Info Model Change	The *Server* supports ModelChange *Event* and NodeVersion *Property* for all *Nodes* that the server allows Model changes for.
Server	Base Info Placeholder Modelling Rules	The *Server* supports defining custom *Object* or *Variables* that include the use of OptionalPlaceholder or MandatoryPlaceholder modelling rules.
Server	Base Info SemanticChange	The *Server* supports SemanticChangeEvent for some Properties. This includes setting the SemanticChange Bit in the status when a semantic change occurs, such as a change in the engineering unit associated with a value.
Server	Base Info EventQueueOverflow EventType	The *Server* supports the EventQueueOverflowEventType as defined in IEC 62541-4.
Server	Base Info OptionSet	The *Server* supports the *VariableType* OptionSet.
Server	Base Info ValueAsText	The *Server* supports the *Property* ValueAsText for enumerated DataTypes.
Server	Base Info Engineering Units	The *Server* supports defining *Variables* that include the Engineering Units *Property*. This property makes use of the EUInformation data structure. This structure by default represents the UN/CEFACT "Codes for Units of Measurement". If a different EU representation is required then the EUInformation.namespaceUri will indicate the alternate namespace.

Category	Title	Description
Server	Base Info Selection List	The *Server* supports Variables of the SelectionListType VariableType.
Server	Base Info FileType Base	The *Server* supports the FileType *Object* (see IEC 62541-5). File writing may be restricted.
Server	Base Info FileType Write	The *Server* supports the FileType *Object*, including writing of files. Also included is the support of user access control on FileType *Object*.
Server	Base Info RequestServerStateChange *Method*	The *Server* supports the RequestServerStateChange *Method*.
Server	Base Info State Machine Instance	Support instances of the StateMachineType or a sub-type in the *AddressSpace*. Generate Events when significant state changes occur. At least one GeneratesEvent *Reference* exists to define the *Event*(s) triggered on state changes.
Server	Base Info Finite State Machine Instance	Support instances of the FiniteStateMachineType or a sub-type in the *AddressSpace*.
Server	Base Info Available States and Transitions	Support the Properties AvailableStates and AvailableTransitions defined for the FiniteStateMachineType.
Client	Base Info *Client* Basic	The *Client* uses the defined OPC UA *AddressSpace*. Access or provide access to *Server* information like the *Server*'s state, BuildInfo, capabilities, Namespace Table and Type Model.
Client	Base Info *Client* Honour Operation Limits	The *Client* shall honour *Server* limits described in ServerCapabilites *Object* of *Server*.
Client	Base Info *Event* Processing	The *Client* is able to subscribe for and process base OPC UA Events.
Client	Base Info *Client* System Status	The *Client* makes use of SystemStatusChangeEventType to detect server shutdowns.
Client	Base Info *Client* Estimated Return Time	*Client* uses the EstimatedReturnTime *Property* to govern the reconnect logic.
Client	Base Info *Client* System Status Underlying System	The *Client* makes use of SystemStatusChangeEventType to detect changes to an Underlying System (SourceNode = *Server*).
Client	Base Info *Client* Device Failure	The *Client* makes use of DeviceFailureEventType to detect failed devices in underlying systems
Client	Base Info *Client* Progress Events	The *Client* makes use of ProgressEvents, including checking for their support.
Client	Base Info *Client* Diagnostics	The *Client* provides interactive or programmatic access to the *Server*'s diagnostic information.
Client	Base Info *Client* Type Programming	The *Client* programmatically process instances of *Objects* or *Variables* by using their type definitions. This includes custom DataTypes, *ObjectTypes* and VariableTypes.
Client	Base Info *Client* Type Pre-Knowledge	The *Client* shall interoperate with *Servers* that do not expose OPC UA Types in *AddressSpace*.
Client	Base Info *Client* Remote *Nodes*	The *Client* can access Nodes that have an extended NodeID that reference a *Server* different then the orginating *Server*. It is acceptable that the *Server* configuration information be pre-configured on the *Client*.
Client	Base Info *Client* Change Events	The *Client* processes ModelChangeEvents to detect changes in the *Server*'s OPC UA *AddressSpace* and take appropriate action for a given change.
Client	Base Info *Client* GetMonitoredItems *Method*	The *Client* makes use of GetMonitoredItems *Method* to recover for communication interruptions and/or to recover subscription information.
Client	Base Data *Client* ResendData *Method*	The *Client* makes use of ResendData *Method* to fetch the last value of the data monitored items.
Client	Base Info *Client* Selection List	The *Client* uses and understands Variables of the SelectionListType VariableType.
Client	Base Info *Client* FileType Base	The *Client* can access a FileType *Object* to transfer a file from the *Server* to the *Client*. This includes large files.
Client	Base Info *Client* FileType Write	The *Client* can access a FileType *Object* to transfer a file from the *Client* to the *Server*. This includes large files.

Category	Title	Description
Client	Base Info *Client* RequestServerStateChange	The *Client* can invoke the RequestServerStateChange *Method*.
Client	Base Info *Client* State Machine Instance	Use instances of the StateMachineType or a sub-type. Monitor either the CurrentState component *Variable* of the instance or the Events triggered as effect of state changes. Use Methods when defined for the StateMachineType to affect the state.
Client	Base Info *Client* Finite State Machine Instance	Use instances of the FinitStateMachineType or a sub-type. Monitor either the CurrentState component *Variable* of the instance or the *Events* triggered as effect of state changes.
Client	Base Info Client Available States and Transitions	Use the Properties AvailableStates and AvailableTransitions when exposed by a *Server*.

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

Table 14 – Address Space Model

Category	Title	Description
Server	Address Space Base	Support the *NodeClasses* with their *Attributes* and *References* as defined in IEC 62541-3. This includes for instance: *Object*, *ObjectType*, *Variable*, *VariableType*, *References* and DataType.
Server	Address Space Atomicity	Support setting the NonatomicRead and NonatomicWrite flags in the AccessLevelEx *Attribute* for *Variable Nodes* to indicate whether Read or Write operations can be performed in atomic manner. If the flags are set to '1', atomicity cannot be assured.
Server	Address Space Full Array Only	Support setting the WriteFullArrayOnly flag in the AccessLevelEx *Attribute* for *Variable Nodes* of non-scalar data types to indicate whether write operations for an array can be performed with an IndexRange.
Server	Address Space Events	Support OPC UA *AddressSpace* elements for generating *Event* notifications. This includes at least one *Node* with an *EventNotifier Attribute* set to True (*Server Node*).
Server	Address Space Complex Data Dictionary	Support structured DataTypes with a Data Dictionary. Note that V1.04 of IEC 62541-3 specifies a simplified approach using the new DataTypeDefinition Attribute. The "Address Space DataTypeDefinition Attribute" Conformance Unit requires support of the DataTypeDefinition Attribute. Support of a DataDictionary will be deprecated in one of the next OPC UA versions.
Server	Address Space DataTypeDefinition Attribute	Support structured DataTypes and expose the meta data and encoding information with a StructureDefinitionType via the DataTypeDefinition *Attribute*.
Server	Address Space *Method*	Support *Method Nodes*.
Server	Address Space Notifier Hierarchy	Supports using the HasNotifier reference to build a hierarchy of *Object Nodes* that are notifiers with other notifier *Object Nodes*.
Server	Address Space Source Hierarchy	Supports hierarchies of event sources where each hierarchy roots in an *Object Node* that is a notifier. The HasEventSource *Reference* is used to relate the *Nodes* within a hierarchy. If *Conditions* are supported, the hierarchy shall include HasCondition *References*.
Server	Address Space WriteMask	Supports WriteMask indicating the write access availability for all attributes, including not supported attributes.
Server	Address Space UserWriteMask	Supports UserWriteMask indicating the write access availability for all attributes for the given user, including not supported attributes. Support includes at least two levels of users.
Server	Address Space UserWriteMask Multilevel	Supports UserWriteMask indicating the write access availability for all attributes for the given user, including not supported attributes. This includes supporting multiple levels of access control for all nodes in the system.
Server	Address Space User Access Level Full	Implements User Access Level security, this includes supporting multiple levels of access control for *Variable* nodes in the system. This includes an indication of read, write, Historical read and Historical write access to the Value *Attribute*.

Category	Title	Description
Server	Address Space User Access Level Base	Implements User Access Level Security for *Variable* nodes, this includes at least two users in the system. This includes an indication of read, write, historical read and Historical write access to the value attribute
Client	Address Space *Client* Base	Uses and understands the *NodeClasses* with their *Attributes* and behaviour as defined in IEC 62541-3. This includes for instance: *Object*, *ObjectType*, *Variable*, *VariableType*, *References* and DataType. This includes treating BrowseNames and String NodeIds as case sensitive.
Client	Address Space *Client* Atomicity	Access the NonatomicRead or NonatomicWrite flags in the AccessLevelEx *Attribute* of *Variable Nodes* to determine whether Read or Write operations can be performed in atomic manner. This information will typically be shown to a user for further action.
Client	Address Space *Client* Full Array Only	Access the WriteFullArrayOnly flag in the AccessLevelEx *Attribute* of *Variable Nodes* with non-scalar data types to determine whether writing to an array with an IndexRange is allowed.
Client	Address Space *Client* Complex Data Dictionary	Uses and understands arbitrary structured DataTypes via Data Dictionary. Note that V1.04 of IEC 62541-3 specifies a simplified approach using the new DataTypeDefinition Attribute. The "Address Space Client DataTypeDefinition Attribute" Conformance Unit requires support of the DataTypeDefinition Attribute.
Client	Address Space *Client* DataTypeDefinition Attribute	Uses and understands arbitrary structured DataTypes where the meta data and encoding information are exposed with the StructureDefinitionType via the DataTypeDefinition *Attribute*.
Client	Address Space *Client* Notifier Hierarchy	Uses hierarchy of *Object Nodes* that are notifiers to detect specific areas where the *Client* can subscribe for Events.
Client	Address Space *Client* Source Hierarchy	Detect and use the hierarchy of event sources exposed for specific *Object Nodes* that are event notifiers.

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

Table 15 – Data Access

Category	Title	Description
Server	Data Access DataItems	Provide *Variables* of DataItemType or one of its subtypes. Support the StatusCodes specified in IEC 62541-8. Support of optional Properties (e.g. "InstrumentRange") shall be verified during certification testing and will be shown in the *Certificate*.
Server	Data Access AnalogItems	Support AnalogItemType *Variables* with corresponding Properties. The support of optional properties will be listed.
Server	Data Access PercentDeadband	Support PercentDeadband filter when monitoring AnalogItemType *Variables*.
Server	Data Access Semantic Changes	Support semantic changes of AnalogItemType items (EURange *Property* and/or EngineeringUnits *Property*). Support semantic change StatusCode bits where appropriate.
Server	Data Access TwoState	Support TwoStateDiscreteType *Variables* with corresponding Properties.
Server	Data Access MultiState	Support MultiStateDiscreteType *Variables* with corresponding Properties.
Server	Data Access MultiStateValueDiscrete	Support MultiStateValueDiscreteType *Variables* with corresponding Properties.
Server	Data Access ArrayItemType	Provide *Variables* of ArrayItemType or one of its subtypes (YArrayItemType, XYArrayItemType, ImageArrayType, CubeArrayType and NDimensionArrayType). The supported subtypes will be listed. Support for this type includes supporting all of the mandatory properties including AxisInformation.
Server	Data Access Complex Number	Supports the Complex Number data type. This data type is available for any variable types that do not have other explicit restrictions.
Server	Data Access DoubleComplex Number	Supports the DoubleComplex Number data type. This data type is available for any variable types that do not have other explicit restrictions.

Category	Title	Description
Client	Data Access *Client* Basic	Understand the DataAccess *Variable* Types. Make use of the standard Properties if applicable.
Client	Data Access *Client* AnalogItems	Understand AnalogItemType *Variables* with corresponding Properties.
Client	Data Access *Client* TwoState	Understand TwoStateDiscreteType *Variables* with corresponding Properties.
Client	Data Access *Client* MultiState	Understand MultiStateDiscreteType *Variables* with corresponding Properties.
Client	Data Access *Client* MultiStateValueDiscrete	Understand MultiStateValueDiscreteType *Variables* with corresponding Properties.
Client	Data Access *Client* Deadband	Use PercentDeadband to filter value changes of AnalogItemType *Variables*.
Client	Data Access *Client* SemanticChange	Recognize the semantic change bit in the StatusCode while monitoring items and take proper action. Typically, the *Client* has to re-read Properties that define type-specific semantic like the EURange and EngineeringUnits Properties.

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

Table 16 – Alarms and Conditions

Category	Title	Description
Server	A & C Basic	Supports *Alarm* & *Condition* model ConditionType.
Server	A & C Enable	Supports Enable and Disable Methods.
Server	A & C Refresh	Supports ConditionRefresh *Method* and the concept of a refresh.
Server	A & C Refresh2	Supports ConditionRefresh2 *Method* and the concept of a monitored item-based refresh.
Server	A & C Instances	Support exposing of A&C *Condition* instances in the *AddressSpace*.
Server	A & C ConditionClasses	Supports multiple *Condition* classes for grouping and filtering of *Alarms*.
Server	A & C *Condition* Sub-Classes	Support assigning multiple *Condition* sub-classes for grouping and filtering of *Alarms*.
Server	A & C Acknowledge	Supports the Acknowledge concept, Acknowledge *Method*, and AcknowledgeableCondition Type.
Server	A & C Confirm	Supports the concept of Confirm and the Confirm *Method*.
Server	A & C Comment	Supports the concept of Comments and the AddComment *Method*.
Server	A & C *Alarm*	Supports the mandatory features of the AlarmCondition Type.
Server	A & C *Alarm* Metrics	Support the collection of alarm metrics data as defined in IEC 62541-9. This implies one of more instances of the AlarmMetricsType.
Server	A & C Branch	Support for branching of *Condition* Types and any subtypes, such as AcknowledgeableConditionType and AlarmConditionType etc.
Server	A & C Shelving	Support the concept of shelving and the TimedShelve, OneShotShelve and Unshelve Methods.
Server	A & C Suppression	Support the SuppressedState.
Server	A & C Suppression by Operator	Support the Suppress and UnSuppress *Methods* to allow an operator control over the SuppressedState.
Server	A & C Silencing	Support the concept of silencing and the Silence Method.
Server	A & C Out Of Service	Support the OutOfService state and the OutOfService Method.
Server	A & C On-Off Delay	Support the OnDelay and OffDelay Properties to eliminate nuisance *Alarms*.
Server	A & C Re-Alarming	Support the ReAlarmTime and ReAlarmRepeatCount Properties that define automatic re-annunciation of *Alarms* in certain conditions.

Category	Title	Description
Server	A & C First in Group *Alarm*	Support the "FirstInGroup" elements for an *Alarm*, indicating which *Alarm* of a group was the trigger.
Server	A & C Audible Sound	Support the AudibleSound *Property*. This *Property* contains the sound file that is to be played if an audible *Alarm* is to be generated.
Server	A & C Exclusive Level	Supports Exclusive Level *Alarm* type.
Server	A & C Exclusive Limit	Supports Exclusive Limit *Alarms*. A *Server* that supports this shall support at least one of the sub-types: Level, Deviation or RateofChange.
Server	A & C Exclusive Deviation	Supports Exclusive Deviation *Alarm* type.
Server	A & C Exclusive RateOfChange	Supports Exclusive RateOfChange *Alarm* type.
Server	A & C Non-Exclusive Limit	Supports Non-Exclusive Limit *Alarms*. A *Server* that supports this shall support at least one of the sub-types: Level, Deviation or RateofChange.
Server	A & C Non-Exclusive Level	Supports Non-Exclusive Level *Alarm* type.
Server	A & C Non-Exclusive Deviation	Supports Non-Exclusive Deviation *Alarm* type.
Server	A & C Non-Exclusive RateOfChange	Supports Non-Exclusive RateOfChange *Alarm* type.
Server	A & C Discrete	Supports Discrete *Alarm* types.
Server	A & C OffNormal	Supports OffNormalAlarmType.
Server	A & C SystemOffNormal	Supports SystemOffNormalAlarmType.
Server	A & C Trip	Supports Trip *Alarm* type.
Server	A & C Discrepancy	Supports Discrepancy *Alarm* type.
Server	A & C Dialog	Supports DialogConditionType including Respond *Method*.
Server	A & C CertificateExpiration	Supports CertificateExpirationAlarmType.
Server	A & E Wrapper Mapping	The *Server* uses the COM A&E mapping specified in the annex of IEC 62541-9 to map OPC-COM Events to A&C Events. This includes *Condition* Class mapping.
Client	A & C Basic *Client*	Uses the *Alarm* & *Condition* model ConditionType.
Client	A & C Enable *Client*	Uses Enable and Disable Methods.
Client	A & C Refresh *Client*	Uses ConditionRefresh *Method* and the concept of a refresh.
Client	A & C Refresh2 *Client*	Uses ConditionRefresh2 *Method* and the concept of a monitored item-based refresh.
Client	A & C Instances *Client*	Uses A&C *Condition* instances when they are exposed in the *AddressSpace*.
Client	A & C ConditionClasses *Client*	Uses *Condition* classes to group *Alarms*.
Client	A & C *Condition* Sub-Classes *Client*	Uses *Condition* sub-classes to group or filter *Alarms*.
Client	A & C Acknowledge *Client*	Understands the Acknowledge concept and the AcknowledgeableCondition Type, and uses the Acknowledge *Method* if requested.
Client	A & C Confirm *Client*	Understands the concept of confirming *Conditions* and uses the Confirm *Method*.
Client	A & C Comment *Client*	Understands the concept of Comments and uses the AddComment *Method*.
Client	A & C *Alarm Client*	Understands the concept of *Alarms* and uses the mandatory features of the AlarmCondition Type,
Client	A & C *Alarm* Metrics *Client*	Understand and use *Alarm* metrics data as defined in IEC 62541-9. This implies discovery of instances of the AlarmMetricsType that can exist anywhere in the HasNotifier hierarchy.
Client	A & C Branch *Client*	Can make use of and process *Condition* Branches, including all actions associated with previous *Condition* instances.
Client	A & C Shelving *Client*	Understand the shelving model and use the TimedShelve, OneShotShelve and Unshelve Methods.
Client	A & C Suppression Client	Understand the SuppressedState model.
Client	A & C Suppression by Operator *Client*	Use the Suppress and UnSuppress *Methods* to allow an operator control over the SuppressedState.
Client	A & C Silencing Client	Understand the SilencedState model and use the Silence Method.

Category	Title	Description
Client	A & C Out Of Service Client	Understand the OutOfServiceState model and use the OutOfService Method.
Client	A & C On-Off Delay *Client*	Uses the OnDelay and OffDelay Properties to eliminate nuisance *Alarms*.
Client	A & C Re-Alarming *Client*	Understand and use the ReAlarmTime and ReAlarmRepeatCount Properties. Configure the ReAlarmTime *Property* for automatic re-annunciation of an *Alarm*. Note that configuration is only possible for *Servers* that expose *Alarm* instances.
Client	A & C First in Group *Alarm Client*	Use the "FirstInGroup" elements for an *Alarm* to determine which *Alarm* of a group was the trigger.
Client	A & C Audible Sound *Client*	Use the AudibleSound *Property* and – if reported – play the sound file.
Client	A & C Exclusive Level *Client*	Uses Exclusive Level *Alarms*.
Client	A & C Exclusive Limit *Client*	Uses Exclusive Limit *Alarms*. Requires that at least one of the sub-types be used.
Client	A & C Exclusive Deviation *Client*	Uses Exclusive Deviation *Alarms*.
Client	A & C Exclusive RateOfChange *Client*	Uses Exclusive RateOfChange *Alarms*.
Client	A & C Non-Exclusive Level *Client*	Uses Non-Exclusive Level *Alarms*.
Client	A & C Non-Exclusive Limit *Client*	Uses Non-Exclusive Limit *Alarms*. Requires that at least one of the sub-types be used.
Client	A & C Non-Exclusive Deviation *Client*	Uses Non-Exclusive Deviation *Alarms*.
Client	A & C Non-Exclusive RateOfChange *Client*	Uses Non-Exclusive RateOfChange *Alarms*.
Client	A & C Discrete *Client*	Uses Discrete *Alarm* types.
Client	A & C OffNormal *Client*	Uses OffNormalAlarmType.
Client	A & C SystemOffNormal *Client*	Uses SystemOffNormalAlarmType.
Client	A & C Trip *Client*	Uses TripAlarmType.
Client	A & C Discrepancy Client	Uses Discrepancy *Alarm* type.
Client	A & C Dialog *Client*	Uses DialogConditionType including Respond *Method*.
Client	A & C CertificateExpiration *Client*	Uses CertificateExpirationAlarmType.

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

Table 17 – Historical Access

Category	Title	Description
Server	Historical Access Read Raw	General support for basic historical access, reading raw data using the ReadRawModifiedDetails structure. Where the time range is specified using a start time, stop time and number of values (a minimum of two of the three parameters shall be provided) and the ReadModified flag is set to False.
Server	Historical Access Data Max *Nodes* Read Continuation Point	Supports enough continuation points to cover the number of supported points indicated in the MaxNodesPerHistoryReadData *Server* OperationLimits *Property* for historical data access.
Server	Historical Access Time Instance	Supports reading historical data at a specified instance in time using the ReadAtTimeDetails structure.
Server	Historical Access Aggregates	Supports reading one or more Aggregates of historical values of *Variables* using the ReadProcessedDetails structure. At least one of the Aggregates described in IEC 62541-13 shall be supported.
Server	Historical Access Insert Value	Supports inserting historical values of *Variables*.
Server	Historical Access Delete Value	Supports deleting historical values of *Variables*.
Server	Historical Access Update Value	Supports updating historical values of *Variables*.
Server	Historical Access Replace Value	Supports replacing historical values of *Variables*.
Server	Historical Access Modified Values	Supports maintaining old values for historical data that have been updated and the retrieval of these values using the ReadRawModifiedDetails structure (ReadModified flag set to true).
Server	Historical Access Annotations	Supports the entry and retrieval of Annotations for historical data. The retrieval is accomplished using the standard historical read raw functionality (ReadRawModifiedDetails). The entry uses the standard historical update (UpdateStructureDataDetails) functionality.
Server	Historical Access ServerTimestamp	Supports providing a ServerTimestamp (as well as the default SourceTimestamp).
Server	Historical Access Structured Data Read Raw	Supports ReadRawModified historical access for structured data. Supporting the structure for an annotation is not considered supporting generic structured data.
Server	Historical Access Structured Data Time Instance	Supports historical access for structured data. Supporting ReadAtTimeDetails for structured data. Supporting the structure for an annotation is not considered supporting generic structured data.
Server	Historical Access Structured Data Insert	Supports historical access for structured data. Inserting Structured data. Supporting the structure for an annotation is not considered supporting generic structured data.
Server	Historical Access Structured Data Delete	Supports historical access for structured data. Delete of existing data. Supporting the structure for an annotation is not considered supporting generic structured data.
Server	Historical Access Structured Data Update	Supports historical access for structured data. Updates of existing data. Supporting the structure for an annotation is not considered supporting generic structured data.
Server	Historical Access Structured Data Replace	Supports replacing structured historical data. Supporting the structure for an annotation is not considered supporting generic structured data.
Server	Historical Access Structured Data Read Modified	Supports maintaining old values for historical structured data that have been updated and the retrieval of these values. Using the ReadRawModifiedDetails structure (ReadModified flag set to true) for structured data. Supporting the structure for an annotation is not considered supporting generic structured data.
Server	Historical Access Events	Supports the retrieval of historical Events using the ReadEventDetails structure. This includes support for simple filtering of Events. The *Event* fields that are stored are server specific, but at least the mandatory fields of BaseEventType are required.

Category	Title	Description
Server	Historical Access *Event* Max Events Read Continuation Point	Supports enough continuation points to cover the number of supported *Event* reads indicated in the MaxNodesPerHistoryReadEvents *Server* OperationLimits *Property* for Historical *Event* access.
Server	Historical Access Insert *Event*	Supports inserting historical Events.
Server	Historical Access Update *Event*	Supports updating historical Events.
Server	Historical Access Replace *Event*	Supports replacing historical Events.
Server	Historical Access Delete *Event*	Supports deleting of historical Events.
Client	Historical Access *Client* Browse	Uses the View *Service* Set to discover *Nodes* with historical data.
Client	Historical Access *Client* Read Raw	Uses the HistoryRead *Service* to read raw historical data using the ReadRawModifiedDetails Structure (ReadModified Flag set to False).
Client	Historical Access *Client* Read Modified	Uses the HistoryRead *Service* to read modified historical data using the ReadRawModifiedDetails Structure (ReadModified Flag set to True).
Client	Historical Access *Client* Read Aggregates	Uses the HistoryRead *Service* to read Aggregated historical data. This includes using at least one of the Aggregates defined in IEC 62541-13.
Client	Historical Access *Client* Structure Data Raw	Uses the HistoryRead *Service* to read raw historical data using the ReadRawModifiedDetails Structure (ReadModified Flag set to False) for structured data.
Client	Historical Access *Client* Structure Data Read Modified	Uses the HistoryRead *Service* to read modified structured historical data using the ReadRawModifiedDetails Structure (ReadModified Flag set to True).
Client	Historical Access *Client* Structure Data Insert	Uses the HistoryUpdate *Service* to insert historical data values for structured data.
Client	Historical Access *Client* Structure Data Delete	Uses the HistoryUpdate *Service* to delete historical data values for structured data.
Client	Historical Access *Client* Structure Data Update	Uses the HistoryUpdate *Service* to update historical data values for structured data.
Client	Historical Access *Client* Structure Data Replace	Uses the HistoryUpdate *Service* to replace historical data values for structured data.
Client	Historical Access *Client* Structure Data Time Instance	Reads historical data at a specified instance in time for structured data. Using the ReadAtTimeDetails structure.
Client	Historical Access *Client* Read Events	Uses the HistoryRead *Service* to read historical *Event* data using the ReadEventDetails Structure.
Client	Historical Access *Client Event* Inserts	Uses the HistoryUpdate *Service* to insert historical Events.
Client	Historical Access *Client Event* Updates	Uses the HistoryUpdate *Service* to update historical Events.
Client	Historical Access *Client Event* Replaces	Uses the HistoryUpdate *Service* to replace historical Events.
Client	Historical Access *Client Event* Deletes	Uses the HistoryUpdate *Service* to delete historical Events.
Client	Historical Access *Client* Data Insert	Uses the HistoryUpdate *Service* to insert historical data values.
Client	Historical Access *Client* Data Delete	Uses the HistoryUpdate *Service* to delete historical data values.
Client	Historical Access *Client* Data Update	Uses the HistoryUpdate *Service* to update historical data values.

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

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

Table 18 – Aggregates

Category	Title	Description
Server	Aggregate Master Configuration	Supports an AggregateConfigurationType *Object* as part of the HistoricalServerCapabilities (defined in IEC 62541-11).
Server	Aggregate Historical Configuration	Supports at least one AggregateConfigurationType *Object*. AggregateConfigurationType *Objects* occur as part of an HistoricalConfiguration *Object*, allowing *Variable* specific configurations.
Server	Aggregate – Interpolative	Supports the Interpolative Aggregate for Historical access.
Server	Aggregate – Average	Supports the Average Aggregate for Historical access.
Server	Aggregate – TimeAverage	Supports the TimeAverage Aggregate for Historical access.
Server	Aggregate – TimeAverage2	Supports the TimeAverage2 Aggregate for Historical access.
Server	Aggregate – Total	Supports the Total Aggregate for Historical access.
Server	Aggregate – Total2	Supports the Total2 Aggregate for Historical access.
Server	Aggregate – Minimum	Supports the Minimum Aggregate for Historical access.
Server	Aggregate – MinimumActualTime	Supports the MinimumActualTime Aggregate for Historical access.
Server	Aggregate – Minimum2	Supports the Minimum2 Aggregate for Historical access.
Server	Aggregate – MinimumActualTime2	Supports the MinimumActualTime2 Aggregate for Historical access.
Server	Aggregate – Maximum	Supports the Maximum Aggregate for Historical access.
Server	Aggregate – MaximumActualTime	Supports the MaximumActualTime Aggregate for Historical access.
Server	Aggregate – Maximum2	Supports the Maximum2 Aggregate for Historical access.
Server	Aggregate – MaximumActualTime2	Supports the MaximumActualTime2 Aggregate for Historical access.
Server	Aggregate – Range	Supports the Range Aggregate for Historical access.
Server	Aggregate – Range2	Supports the Range2 Aggregate for Historical access.
Server	Aggregate – Count	Supports the Count Aggregate for Historical access.
Server	Aggregate – DurationInStateZero	Supports the DurationInStateZero Aggregate for Historical access.
Server	Aggregate – DurationInStateNonZero	Supports the DurationInStateNonZero Aggregate for Historical access.
Server	Aggregate – NumberOfTransitions	Supports the NumberOfTransitions Aggregate for Historical access.
Server	Aggregate – Start	Supports the Start Aggregate for Historical access.
Server	Aggregate – StartBound	Supports the StartBound Aggregate for Historical access.
Server	Aggregate – End	Supports the End Aggregate for Historical access.
Server	Aggregate – EndBound	Supports the EndBound Aggregate for Historical access.
Server	Aggregate – Delta	Supports the Delta Aggregate for Historical access.
Server	Aggregate – DeltaBounds	Supports the DeltaBounds Aggregate for Historical access.
Server	Aggregate – DurationGood	Supports the DurationGood Aggregate for Historical access.
Server	Aggregate – DurationBad	Supports the DurationBad Aggregate for Historical access.
Server	Aggregate – PercentGood	Supports the PercentGood Aggregate for Historical access.
Server	Aggregate – PercentBad	Supports the PercentBad Aggregate for Historical access.
Server	Aggregate – WorstQuality	Supports the WorstQuality Aggregate for Historical access.
Server	Aggregate – WorstQuality2	Supports the WorstQuality2 Aggregate for Historical access.
Server	Aggregate – AnnotationCount	Supports the AnnotationCount Aggregate for Historical access.
Server	Aggregate – StandardDeviationSample	Supports the StandardDeviationSample Aggregate for Historical access.
Server	Aggregate – VarianceSample	Supports the VarianceSample Aggregate for Historical access.

Category	Title	Description
Server	Aggregate – StandardDeviationPopulation	Supports the StandardDeviationPopulation for Historical access.
Server	Aggregate – VariancePopulation	Supports the VariancePopulation for Historical access.
Server	Aggregate – Custom	The *Server* supports custom Aggregates for Historical access that do not have standard tests defined. These Aggregates are list as untested by this *ConformanceUnit*.
Server	Aggregate *Subscription* – Filter	Supports Aggregate subscription filters which requires at least one of the defined Aggregates is supported as defined in IEC 62541-13.
Server	Aggregate *Subscription* – Interpolative	Supports subscription filter for the Interpolative Aggregate.
Server	Aggregate *Subscription* – Average	Supports subscription filter for the Average Aggregate.
Server	Aggregate *Subscription* – TimeAverage	Supports subscription filter for the TimeAverage Aggregate.
Server	Aggregate *Subscription* – TimeAverage2	Supports subscription filter for the TimeAverage2 Aggregate.
Server	Aggregate *Subscription* – Total	Supports subscription filter for the Total Aggregate.
Server	Aggregate *Subscription* – Total2	Supports subscription filter for the Total2 Aggregate.
Server	Aggregate *Subscription* – Minimum	Supports subscription filter for the Minimum Aggregate.
Server	Aggregate *Subscription* – MinimumActualTime	Supports subscription filter for the MinimumActualTime Aggregate.
Server	Aggregate *Subscription* – Minimum2	Supports subscription filter for the Minimum2 Aggregate.
Server	Aggregate *Subscription* – MinimumActualTime2	Supports subscription filter for the MinimumActualTime2 Aggregate.
Server	Aggregate *Subscription* – Maximum	Supports subscription filter for the Maximum Aggregate.
Server	Aggregate *Subscription* – MaximumActualTime	Supports subscription filter for the MaximumActualTime Aggregate.
Server	Aggregate *Subscription* – Maximum2	Supports subscription filter for the Maximum2 Aggregate.
Server	Aggregate *Subscription* – MaximumActualTime2	Supports subscription filter for the MaximumActualTime2 Aggregate.
Server	Aggregate *Subscription* – Range	Supports subscription filter for the Range Aggregate.
Server	Aggregate *Subscription* – Range2	Supports subscription filter for the Range2 Aggregate.
Server	Aggregate *Subscription* – Count	Supports subscription filter for the Count Aggregate.
Server	Aggregate *Subscription* – DurationInStateZero	Supports subscription filter for the DurationInStateZero Aggregate.
Server	Aggregate *Subscription* – DurationInStateNonZero	Supports subscription filter for the DurationInStateNonZero Aggregate.
Server	Aggregate *Subscription* – NumberOfTransitions	Supports subscription filter for the NumberOfTransitions Aggregate.
Server	Aggregate *Subscription* – Start	Supports subscription filter for the Start Aggregate.
Server	Aggregate *Subscription* – StartBound	Supports subscription filter for the StartBound Aggregate.
Server	Aggregate *Subscription* – End	Supports subscription filter for the End Aggregate.
Server	Aggregate *Subscription* – EndBound	Supports subscription filter for the EndBound Aggregate.
Server	Aggregate *Subscription* – Delta	Supports subscription filter for the Delta Aggregate.
Server	Aggregate *Subscription* – DeltaBounds	Supports subscription filter for the DeltaBounds Aggregate.

Category	Title	Description
Server	Aggregate *Subscription* – DurationGood	Supports subscription filter for the DurationGood Aggregate.
Server	Aggregate *Subscription* – DurationBad	Supports subscription filter for the DurationBad Aggregate.
Server	Aggregate *Subscription* – PercentGood	Supports subscription filter for the PercentGood Aggregate.
Server	Aggregate *Subscription* – PercentBad	Supports subscription filter for the PercentBad Aggregate.
Server	Aggregate *Subscription* – WorstQuality	Supports subscription filter for the WorstQuality Aggregate.
Server	Aggregate *Subscription* – WorstQuality2	Supports subscription filter for the WorstQuality2 Aggregate.
Server	Aggregate *Subscription* – AnnotationCount	Supports subscription filter for the AnnotationCount Aggregate.
Server	Aggregate *Subscription* – StandardDeviationSample	Supports subscription filter for the StandardDeviationSample Aggregate.
Server	Aggregate *Subscription* – VarianceSample	Supports subscription filter for the VarianceSample Aggregate.
Server	Aggregate *Subscription* – StandardDeviationPopulation	Supports subscription filter for the StandardDeviationPopulation Aggregate.
Server	Aggregate *Subscription* – VariancePopulation	Supports subscription filter for the VariancePopulation Aggregate.
Server	Aggregate *Subscription* – Custom	The *Server* supports subscribing to custom Aggregates that do not have standard tests defined. These Aggregates are listed as untested by this *ConformanceUnit*.
Client	Aggregate – *Client* Usage	Uses Historical access to Aggregate which requires at least one of the defined Aggregates is supported as defined in IEC 62541-13.
Client	Aggregate – *Client* Interpolative	Uses Historical access to the Interpolative Aggregate.
Client	Aggregate – *Client* Average	Uses Historical access to the Average Aggregate.
Client	Aggregate – *Client* TimeAverage	Uses Historical access to the TimeAverage Aggregate.
Client	Aggregate – *Client* TimeAverage2	Uses Historical access to the TimeAverage2 Aggregate.
Client	Aggregate – *Client* Total	Uses Historical access to the Total Aggregate.
Client	Aggregate – *Client* Total2	Uses Historical access to the Total2 Aggregate.
Client	Aggregate – *Client* Minimum	Uses Historical access to the Minimum Aggregate.
Client	Aggregate – *Client* MinimumActualTime	Uses Historical access to the MinimumActualTime Aggregate.
Client	Aggregate – *Client* Minimum2	Uses Historical access to the Minimum2 Aggregate.
Client	Aggregate – *Client* MinimumActualTime2	Uses Historical access to the MinimumActualTime2 Aggregate.
Client	Aggregate – *Client* Maximum	Uses Historical access to the Maximum Aggregate.
Client	Aggregate – *Client* MaximumActualTime	Uses Historical access to the MaximumActualTime Aggregate.
Client	Aggregate – *Client* Maximum2	Uses Historical access to the Maximum2 Aggregate.
Client	Aggregate – *Client* MaximumActualTime2	Uses Historical access to the MaximumActualTime2 Aggregate.
Client	Aggregate – *Client* Range	Uses Historical access to the Range Aggregate.
Client	Aggregate – *Client* Range2	Uses Historical access to the Range2 Aggregate.
Client	Aggregate – *Client* Count	Uses Historical access to the Count Aggregate.
Client	Aggregate – *Client* DurationInStateZero	Uses Historical access to the DurationInStateZero Aggregate.
Client	Aggregate – *Client* DurationInStateNonZero	Uses Historical access to the DurationInStateNonZero Aggregate.

Category	Title	Description
Client	Aggregate – *Client* NumberOfTransitions	Uses Historical access to the NumberOfTransitions Aggregate.
Client	Aggregate – *Client* Start	Uses Historical access to the Start Aggregate.
Client	Aggregate – *Client* StartBound	Uses Historical access to the StartBound Aggregate.
Client	Aggregate – *Client* End	Uses Historical access to the End Aggregate.
Client	Aggregate – *Client* EndBound	Uses Historical access to the EndBound Aggregate.
Client	Aggregate – *Client* Delta	Uses Historical access to the Delta Aggregate.
Client	Aggregate – *Client* DeltaBounds	Uses Historical access to the DeltaBounds Aggregate.
Client	Aggregate – *Client* DurationGood	Uses Historical access to the DurationGood Aggregate.
Client	Aggregate – *Client* DurationBad	Uses Historical access to the DurationBad Aggregate.
Client	Aggregate – *Client* PercentGood	Uses Historical access to the PercentGood Aggregate.
Client	Aggregate – *Client* PercentBad	Uses Historical access to the PercentBad Aggregate.
Client	Aggregate – *Client* WorstQuality	Uses Historical access to the WorstQuality Aggregate.
Client	Aggregate – *Client* WorstQuality2	Uses Historical access to the WorstQuality2 Aggregate.
Client	Aggregate – *Client* AnnotationCount	Uses Historical access to the AnnotationCount Aggregate.
Client	Aggregate – *Client* StandardDeviationSample	Uses Historical access to the StandardDeviationSample Aggregate.
Client	Aggregate – *Client* VarianceSample	Uses Historical access to the VarianceSample Aggregate.
Client	Aggregate – *Client* StandardDeviationPopulation	Uses Historical access to the StandardDeviationPopulation Aggregate.
Client	Aggregate – *Client* VariancePopulation	Uses Historical access to the VariancePopulation Aggregate.
Client	Aggregate – *Client* Custom Aggregates	The *Client* can make use of all custom Aggregates in the list of Aggregates, via Historical access, exposed by the *Server*. This includes displaying or utilizing the data in some manner.
Client	Aggregate *Subscription* – *Client* Filter	Subscribes for data using Aggregate filters which requires at least one of the Aggregates defined in IEC 62541-13 is supported.
Client	Aggregate *Subscription* – *Client* Interpolative	Subscribes for data using the Interpolative Aggregate filter.
Client	Aggregate *Subscription* – *Client* Average	Subscribes for data using the Average Aggregate filter.
Client	Aggregate *Subscription* – *Client* TimeAverage	Subscribes for data using the TimeAverage Aggregate filter.
Client	Aggregate *Subscription* – *Client* TimeAverage2	Subscribes for data using the TimeAverage2 Aggregate filter.
Client	Aggregate *Subscription* – *Client* Total	Subscribes for data using the Total Aggregate filter.
Client	Aggregate *Subscription* – *Client* Total2	Subscribes for data using the Total2 Aggregate filter.
Client	Aggregate *Subscription* – *Client* Minimum	Subscribes for data using the Minimum Aggregate filter.
Client	Aggregate *Subscription* – *Client* MinimumActualTime	Subscribes for data using the MinimumActualTime Aggregate filter.
Client	Aggregate *Subscription* – *Client* Minimum2	Subscribes for data using the Minimum2 Aggregate filter.
Client	Aggregate *Subscription* – *Client* MinimumActualTime2	Subscribes for data using the MinimumActualTime2 Aggregate filter.
Client	Aggregate *Subscription* – *Client* Maximum	Subscribes for data using the Maximum Aggregate filter.
Client	Aggregate *Subscription* – *Client* MaximumActualTime	Subscribes for data using the MaximumActualTime Aggregate filter.

Category	Title	Description
Client	Aggregate *Subscription* – *Client* MaximumActualTime2	Subscribes for data using the MaximumActualTime2 Aggregate filter.
Client	Aggregate *Subscription* – *Client* Maximum2	Subscribes for data using the Maximum2 Aggregate filter.
Client	Aggregate *Subscription* – *Client* Range	Subscribes for data using the Range Aggregate filter.
Client	Aggregate *Subscription* – *Client* Range2	Subscribes for data using the Range2 Aggregate filter.
Client	Aggregate *Subscription* – *Client* Count	Subscribes for data using the Count Aggregate filter.
Client	Aggregate *Subscription* – *Client* DurationInStateZero	Subscribes for data using the DurationInStateZero Aggregate filter.
Client	Aggregate *Subscription* – *Client* DurationInStateNonZero	Subscribes for data using the DurationInStateNonZero Aggregate filter.
Client	Aggregate *Subscription* – *Client* NumberOfTransitions	Subscribes for data using the NumberOfTransitions Aggregate filter.
Client	Aggregate *Subscription* – *Client* Start	Subscribes for data using the Start Aggregate filter.
Client	Aggregate *Subscription* – *Client* StartBound	Subscribes for data using the StartBound Aggregate filter.
Client	Aggregate *Subscription* – *Client* End	Subscribes for data using the End Aggregate filter.
Client	Aggregate *Subscription* – *Client* EndBound	Subscribes for data using the EndBound Aggregate filter.
Client	Aggregate *Subscription* – *Client* Delta	Subscribes for data using the Delta Aggregate filter.
Client	Aggregate *Subscription* – *Client* DeltaBounds	Subscribes for data using the DeltaBounds Aggregate filter.
Client	Aggregate *Subscription* – *Client* DurationGood	Subscribes for data using the DurationGood Aggregate filter.
Client	Aggregate *Subscription* – *Client* DurationBad	Subscribes for data using the DurationBad Aggregate filter.
Client	Aggregate *Subscription* – *Client* PercentGood	Subscribes for data using the PercentGood Aggregate filter.
Client	Aggregate *Subscription* – *Client* PercentBad	Subscribes for data using the PercentBad Aggregate filter.
Client	Aggregate *Subscription* – *Client* WorstQuality	Subscribes for data using the WorstQuality Aggregate filter.
Client	Aggregate *Subscription* – *Client* WorstQuality2	Subscribes for data using the WorstQuality2 Aggregate filter.
Client	Aggregate *Subscription* – *Client* AnnotationCount	Subscribes for data using the AnnotationCount Aggregate filter.
Client	Aggregate *Subscription* – *Client* StandardDeviationSample	Subscribes for data using the StandardDeviationSample Aggregate filter.
Client	Aggregate *Subscription* – *Client* VarianceSample	Subscribes for data using the VarianceSample Aggregate filter.
Client	Aggregate *Subscription* – *Client* StandardDeviationPopulation	Subscribes for data using the StandardDeviationPopulation Aggregate filter.
Client	Aggregate *Subscription* – *Client* VariancePopulation	Subscribes for data using the VariancePopulation Aggregate filter.
Client	Aggregate *Subscription* – *Client* Custom Aggregates	The *Client* supports subscribing to all custom Aggregates in the list of Aggregates exposed by the *Server*. This includes displaying or utilizing the data in some manner.

Table 19 describes auditing related items that can be profiled. Most full function *Servers* would support these features, although some resource constrained *Servers* may not provide this functionality. *Clients* that are security aware or are used to support security logging would support these features.

Table 19 – Auditing

Category	Title	Description
Server	Auditing Base	Support AuditEvents. The list of supported AuditEvents shall be verified during certification testing and will be shown in the certification test result. Base AuditEvents are defined in IEC 62541-3 and in IEC 62541-5.
Client	Auditing *Client* Audit ID	*Client* supports generating AuditEvents ids and providing them to *Servers*.
Client	Auditing *Client* Subscribes	The *Client* supports subscribing for AuditEvents and storing / processing them in a secure manner.

Table 20 describes Redundancy related items that are profiled. *Servers* that support redundancy would support appropriate *ConformanceUnits* based on the type of redundancy they support. *Clients* that are capable of handling redundancy would support the appropriate *ConformanceUnits* based of the type of redundancy they support.

Table 20 – Redundancy

Category	Title	Description
Server	Redundancy *Server*	Supports *Server* based redundancy.
Server	Redundancy *Server* Transparent	Supports transparent *Server* redundancy.
Client	Redundancy *Client*	*Client* supports *Client* redundancy. *Clients* that support *Client* redundancy can failover to another *Client* (requires some out of band communication).
Client	Redundancy *Client* Switch	*Clients* supporting this *ConformanceUnit* monitor the redundancy status for non-transparent redundancy *Servers* and switch to the backup *Server* when they recognize a change in server status.

Table 21 describes items for a Global *Discovery Server* (GDS). *Servers* that act as a GDS would support these *ConformanceUnits*.

Table 21 – Global Discovery Server

Category	Title	Description
Global Directory *Service*	GDS Application Directory	Supports the Directory *Object* with all Methods like RegisterApplication and QueryServers.
Global Directory *Service*	GDS Query Applications	Supports the QueryApplications *Method* on the Directory *Object* specified in IEC 62541-12.
Global Directory *Service*	GDS LDS-ME Connectivity	The GDS can be configured to use specific LDS-ME installations for semi-automatic application registration for all *Servers* on a subnet.
Global Directory *Service*	GDS *Certificate* Manager Pull Model	This Conformance Unit requires support of the complete Information Model and *Services* for *Certificate* management including the Pull Model as specified in IEC 62541-12.
Global Directory *Service*	GDS *Certificate* Manager Push Model	This Conformance Unit requires use of the complete Information model and *Services* for the *Certificate* management Push Model as specified in IEC 62541-12.
Global Directory *Service*	GDS Key Credential *Service* Pull Model	This Conformance Unit requires support of the complete Information Model and *Services* for KeyCredential Pull Management as specified in IEC 62541-12.
Global Directory *Service*	GDS Key Credential *Service* Push Model	This Conformance Unit requires use of the complete Information model and *Services* for KeyCredential Push Management as specified in IEC 62541-12.
Global Directory *Service*	GDS Authorization *Service Server*	This Conformance Unit requires support of AuthorizationServiceType Objects as specified in IEC 62541-12. OA Clients use the RequestAccessToken Method on these Objects to request an Access Token from an Identity Provider.

5.5 Miscellaneous

Table 22 describes miscellaneous *ConformanceUnits*.

Table 22 – Miscellaneous

Category	Title	Description
Server	Documentation – Supported *Profiles*	The documentation includes a description of the profiles supported by the product. This description includes the level of Certification testing the product has passed.
Server	Documentation – Multiple Languages	The documentation is available in multiple languages. The results of this conformance unit include the list of supported languages.
Server	Documentation – Users Guide	The application includes documentation that describes the available functionality provided by the application. For *Servers* it includes a summary of all functionality provided by the *Server*.
Server	Documentation – On-line	The documentation provided by the application is available in electronic format as part of the application. The electronic documentation, could be a WEB page, installed document or CD/DVD, but in all case it can be accessed from the application or from a link installed with the application.
Server	Documentation – Installation	The application includes installation instructions that are sufficient to easily install the application. This includes descriptions of any and all possible configuration items. Instructions for loading or configuring security related items such as Application Instance *Certificates*.
Server	Documentation – Trouble Shooting Guide	The application includes documentation that describes typical problems a user may encounter and actions that the user could perform to resolve the problem. It could also describe tip, tricks or other actions that could help a user diagnose or fix a problem. It could also describe tools or other items that can be used in diagnosing or repairing problems. The actual Trouble Shooting Guide can be part of other documentation, but should be complete enough to provide useful information to a novice user.
Client	Documentation *Client* – Supported *Profiles*	The documentation includes a description of the profiles supported by the product. This description includes any software certificates that describes the level of Certification testing the product has passed.
Client	Documentation *Client* – Multiple Languages	The documentation is available in multiple languages. The results of this conformance unit include the list of supported languages.
Client	Documentation *Client* – Users Guide	The application includes documentation that describes the available functionality provided by the application. For client applications, this includes any operator restrictions or general functionality that the client application makes use of.
Client	Documentation *Client* – On-line	The documentation provided by the application is available in electronic format as part of the application. The electronic documentation could be a WEB page, installed document or CD/DVD, but in all cases it can be accessed from the application or from a link installed with the application.
Client	Documentation *Client* – Installation	The application includes installation instructions that are sufficient to easily install the application. This includes descriptions of any and all possible configuration items. Instructions for loading or configuring security related items such as Application Instance *Certificates*.
Client	Documentation *Client* – Trouble Shooting Guide	The application includes documentation that describes typical problems a user may encounter and actions that the user could perform to resolve the problem. It could also describe tips, tricks or other actions that could help a user diagnose or fix a problem. It could also describe tools or other items that can be used in diagnosing or repairing problems. The actual Trouble Shooting Guide can be part of other documentation, but should be complete enough to provide useful information to a novice user.
Security	Best Practice – Timeouts	The user is able to configure reasonable timeouts for Secure Channels, sessions and subscriptions to limit Denial of *Service* and resource consumption issues (see IEC 62541-2 for additional details).
Security	Best Practice – Strict *Message* Handling	The application assures that messages that are illegally or incorrectly formed are rejected with appropriate error code or appropriate actions as specified in IEC 62541-4 and IEC 62541-6.
Security	Best Practice – Random Numbers	All random numbers that are required for security use appropriate cryptographic library based random number generators.
Security	Best Practice – Administrative Access	The *Server* and *Client* allow for appropriate restriction of access to administrative personnel. This includes multiple levels of administrative access on platforms that support multiple administrative roles (such as Windows or Linux).

Category	Title	Description
Security	Best Practice – *Alarm* Handling	A *Server* should restrict critical alarm functionality to users that have the appropriate rights to perform these actions. This would include disabling or alarms, shelving of alarms and generation of dialog messages. It would also include other security related functionality such maintaining appropriate timeouts for shelving and dialogs and preventing an overload of dialog messages.
Security	Best Practice – Audit Events	Subscriptions for Audit Events are restricted to authorized personnel. A *Server* may also reject a *Subscription* for Audit Events that is not over a Secure Channel if one is available.
Security	Best Practice – Audit Events *Client*	Audit tracking system connects to a *Server* using a Secure Channel and under the appropriate administrative rights to allow access to Audit Events.

6 Profiles

6.1 Overview

This clause includes a listing of the categories that a *Profile* can be grouped into, a list of named *Profiles* and the detailed listing of each *Profile* including directly defined *ConformanceUnits* and any sub *Profiles* that are included in the *Profile*.

6.2 Profile list

Table 23 lists *Profiles*. The *Profile* table is ordered by *Profile* category and then alphabetically by the name of the *Profile*. The table includes a list of categories the *Profile* is associated with and a URI. The URI is used to uniquely identify a *Profile*. The URI shall be able to be used to access the information provided in this document with regard to the given *Profile* in an on-line display.

An application (*Client* or *Server*) shall implement all of the *ConformanceUnits* in a *Profile* in order to be compliant with the *Profile*. Some *Profiles* contain optional *ConformanceUnits*. An optional *ConformanceUnit* means that an application has the option to not support the *ConformanceUnit*. However, if supported, the application shall pass all tests associated with the *ConformanceUnit*. For example, some *ConformanceUnits* require specific information model items to be available. They are, therefore, listed as optional in order to allow for the information model items to be omitted. If a *Server* desires to be listed as supporting the optional *ConformanceUnit* then it shall include any required information model items in the configuration provided for certification testing. The test result that is generated by the certification testing lists all optional *ConformanceUnits* and whether they are supported or not by the tested UA application. Some *ConformanceUnits* also include lists of supported DataTypes or optional Subtypes; the lists are handled in the same manner as optional *ConformanceUnits*. All reporting requirements for optional *ConformanceUnits* also apply to these lists of supported DataTypes or Subtypes.

Table 23 – Profile list

Profile	Related Category	URI
Core 2017 Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/Core2017Facet
Sessionless *Server* Facet	*Server*	http://opcfoundation.org/UA-*Profile*/*Server*/SessionLess
Reverse Connect Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/ReverseConnect
Base Server Behaviour Facet	Server	http://opcfoundation.org/UA-Profile/Server/Behaviour
Request State Change Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/RequestStateChange
Subnet Discovery Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/SubnetDiscovery
Global Certificate Management Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/GlobalCertificateManagement
Authorization Service Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/AuthorizationServiceConfiguration
KeyCredential Service Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/KeyCredentialManagement
Attribute WriteMask Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/AttributeWriteMask
File Access Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/FileAccess
Documentation Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/Documentation
Embedded DataChange Subscription Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/EmbeddedDataChangeSubscription
Standard DataChange Subscription 2017 Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/StandardDataChangeSubscription2017
Enhanced DataChange Subscription 2017 Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/EnhancedDataChangeSubscription2017
Durable Subscription Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/DurableSubscription
Data Access Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/DataAccess
ComplexType 2017 Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/ComplexTypes2017
Standard Event Subscription Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/StandardEventSubscription
Address Space Notifier Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/AddressSpaceNotifier
A & C Base Condition Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/ACBaseCondition
A & C Refresh2 Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/ACRefresh2
A & C Address Space Instance Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/ACAddressSpaceInstance
A & C Enable Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/ACEnable
A & C AlarmMetrics Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/ACAlarmMetrics
A & C Alarm Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/ACAlarm
A & C Acknowledgeable Alarm Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/ACAckAlarm
A & C Exclusive Alarming Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/ACExclusiveAlarming
A & C Non-Exclusive Alarming Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/ACNon-ExclusiveAlarming
A & C Previous Instances Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/ACPreviousInstances
A & C Dialog Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/ACDialog

Profile	Related Category	URI
A & C CertificateExpiration Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/ACCertificateExpiration
A & E Wrapper Facet	Server	http://opcfoundation.org/UA-Profile/Server/AEWrapper
Method Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/Methods
Auditing Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/Auditing
Node Management Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/NodeManagement
User Role Base Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/UserRoleBase
User Role Management Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/UserRoleManagement
State Machine Server Facet	Server	http://opcfoundation.org/UA-*Profile*/*Server*/StateMachine
Client Redundancy Server Facet	Server	http://opcfoundation.org/UA-*Profile*/*Server*/ClientRedundancy
Redundancy Transparent Server Facet	Server	http://opcfoundation.org/UA-*Profile*/*Server*/TransparentRedundancy
Redundancy Visible Server Facet	Server	http://opcfoundation.org/UA-*Profile*/*Server*/VisibleRedundancy
Historical Raw Data Server Facet	Server	http://opcfoundation.org/UA-*Profile*/*Server*/HistoricalRawData
Historical Aggregate Server Facet	Server	http://opcfoundation.org/UA-*Profile*/*Server*/AggregateHistorical
Historical Data AtTime Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/HistoricalDataAtTime
Historical Access Modified Data Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/HistoricalModifiedData
Historical Annotation Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/HistoricalAnnotation
Historical Data Insert Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/HistoricalDataInsert
Historical Data Update Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/HistoricalDataUpdate
Historical Data Replace Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/HistoricalDataReplace
Historical Data Delete Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/HistoricalDataDelete
Historical Access Structured Data Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/HistoricalStructuredData
Base Historical Event Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/BaseHistoricalEvent
Historical Event Update Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/HistoricalEventUpdate
Historical Event Replace Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/HistoricalEventReplace
Historical Event Insert Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/HistoricalEventInsert
Historical Event Delete Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/HistoricalEventDelete
Aggregate Subscription Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/AggregateSubscription
Nano Embedded Device 2017 Server Profile	Server	http://opcfoundation.org/UA-Profile/Server/NanoEmbeddedDevice2017
Micro Embedded Device 2017 Server Profile	Server	http://opcfoundation.org/UA-Profile/Server/MicroEmbeddedDevice2017

Profile	Related Category	URI
Embedded 2017 UA Server Profile	Server	http://opcfoundation.org/UA-Profile/Server/EmbeddedUA2017
Standard 2017 UA Server Profile	Server	http://opcfoundation.org/UA-Profile/Server/StandardUA2017
Core 2017 *Client* Facet	*Client*	http://opcfoundation.org/UA-*Profile*/*Client*/Core2017
Sessionless *Client* Facet	*Client*	http://opcfoundation.org/UA-*Profile*/*Client*/SessionLess
Reverse Connect Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/ReverseConnect
Base Client Behaviour Facet	Client	http://opcfoundation.org/UA-Profile/Client/Behaviour
Discovery Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/Discovery
Subnet Discovery Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/SubnetDiscovery
Global Discovery Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/GlobalDiscovery
Global Certificate Management Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/GlobalCertificateManagement
KeyCredential Service Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/KeyCredentialManagement
Access Token Request Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/AccessTokenRequest
AddressSpace Lookup Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/AddressSpaceLookup
Request State Change Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/RequestStateChange
File Access Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/FileAccess
Entry Level Support 2015 Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/EntryLevelSupport2015
Multi-Server Client Connection Facet	Client	http://opcfoundation.org/UA-Profile/Client/MultiServer
Documentation – Client	Client	http://opcfoundation.org/UA-Profile/Client/Documentation
Attribute Read Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/AttributeRead
Attribute Write Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/AttributeWrite
DataChange Subscriber Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/DataChangeSubscriber
Durable Subscription Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/DurableSubscription
DataAccess Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/DataAccess
Event Subscriber Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/EventSubscriber
Base Event Processing Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/BaseEventProcessing
Notifier and Source Hierarchy Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/NotifierAndSourceHierarchy
A & C Base Condition Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/ACBaseCondition
A & C Refresh2 Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/ACRefresh2
A & C Address Space Instance Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/ACAddressSpaceInstance
A & C Enable Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/ACEnable
A & C AlarmMetrics Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/ACAlarmMetrics
A & C Alarm Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/ACAlarm
A & C Exclusive Alarming Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/ACExclusiveAlarming
A & C Non-Exclusive Alarming Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/ACNon-ExclusiveAlarming

Profile	Related Category	URI
A & C Previous Instances Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/ACPreviousInstances
A & C Dialog Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/ACDialog
A & C CertificateExpiration Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/ACCertificateExpiration
A & E Proxy Facet	Client	http://opcfoundation.org/UA-Profile/Client/AEProxy
Method Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/Method
Auditing Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/Auditing
Node Management Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/NodeManagement
Advanced Type Programming Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/TypeProgramming
User Role Management Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/UserRoleManagement
State Machine Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/StateMachine
Diagnostic Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/Diagnostic
Redundant Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/Redundancy
Redundancy Switch Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/RedundancySwitch
Historical Access Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalAccess
Historical Data AtTime Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalAccessAtTime
Historical Aggregate Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalAccessAggregate
Historical Annotation Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalAnnotation
Historical Access Modified Data Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalAccessModifiedData
Historical Data Insert Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalInsertData
Historical Data Update Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalUpdateData
Historical Data Replace Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalReplaceData
Historical Data Delete Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalDeleteData
Historical Access Client Server Timestamp Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalServerTimeStamp
Historical Structured Data Access Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalAccessStructuredData
Historical Structured Data AtTime Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalAtTimeStructuredData
Historical Structured Data Modified Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalModifiedStructuredData
Historical Structured Data Insert Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalInsertStructuredData
Historical Structured Data Update Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalUpdateStructuredData
Historical Structured Data Replace Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalReplaceStructuredData
Historical Structured Data Delete Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalDeleteStructuredData

Profile	Related Category	URI
Historical Events Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalEvents
Historical Event Insert Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalInsertEvents
Historical Event Update Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalUpdateEvents
Historical Event Replace Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalReplaceEvents
Historical Event Delete Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalDeleteEvents
Aggregate Subscriber Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/AggregateSubscriber
Standard UA Client Profile	Client	http://opcfoundation.org/UA-Profile/Client/Standard
Standard UA Client 2017 Profile	Client	http://opcfoundation.org/UA-Profile/Client/Standard2017
UA-TCP UA-SC UA-Binary	Transport	http://opcfoundation.org/UA-Profile/Transport/uatcp-uasc-uabinary
HTTPS UA-Binary	Transport	http://opcfoundation.org/UA-Profile/Transport/https-uabinary
HTTPS UA-XML	Transport	http://opcfoundation.org/UA-Profile/Transport/https-uasoapxml
HTTPS UA-JSON	Transport	http://opcfoundation.org/UA-Profile/Transport/https-uajson
WSS UA-SC UA-Binary	Transport	http://opcfoundation.org/UA-Profile/Transport/wss-uasc-uabinary
WSS UA-JSON	Transport	http://opcfoundation.org/UA-Profile/Transport/wss-uajson
Security User Access Control Full	Security, Server	http://opcfoundation.org/UA-Profile/Security/UserAccessFull
Security User Access Control Base	Security, Server	http://opcfoundation.org/UA-Profile/Security/UserAccessBase
Security Time Synchronization	Security	http://opcfoundation.org/UA-Profile/Security/TimeSync
Best Practice – Audit Events	Security, Server	http://opcfoundation.org/UA-Profile/Security/BestPracticeAuditEvents
Best Practice – Alarm Handling	Security, Server	http://opcfoundation.org/UA-Profile/Security/BestPracticeAlarmHandling
Best Practice – Random Numbers	Security	http://opcfoundation.org/UA-Profile/Security/BestPracticeRandomNumbers
Best Practice – Timeouts	Security	http://opcfoundation.org/UA-Profile/Security/BestPracticeTimeouts
Best Practice – Administrative Access	Security	http://opcfoundation.org/UA-Profile/Security/BestPracticeAdministrativeAccess
Best Practice – Strict Message Handling	Security, Server	http://opcfoundation.org/UA-Profile/Security/BestPracticeStrictMessage
Best Practice – Audit Events Client	Client, Security	http://opcfoundation.org/UA-Profile/Security/BestPracticeAuditEventsClient
TransportSecurity – TLS 1.2	Security	http://opcfoundation.org/UA-Profile/TransportSecurity/TLS-1-2
TransportSecurity – TLS 1.2 with PFS	Security	http://opcfoundation.org/UA-Profile/TransportSecurity/TLS-1-2-PFS
SecurityPolicy – None	Security	http://opcfoundation.org/UA/SecurityPolicy#None
SecurityPolicy [A] – Aes128-Sha256-RsaOaep	Security	http://opcfoundation.org/UA/SecurityPolicy#Aes128_Sha256_RsaOaep
SecurityPolicy [B] – Basic256Sha256	Security	http://opcfoundation.org/UA/SecurityPolicy#Basic256Sha256

Profile	Related Category	URI
SecurityPolicy – Aes256-Sha256-RsaPss	Security	http://opcfoundation.org/UA/SecurityPolicy#Aes256_Sha256_RsaPss
User Token – Anonymous Facet	Security	http://opcfoundation.org/UA-Profile/Security/UserToken/Anonymous
User Token – User Name Password Server Facet	Security, Server	http://opcfoundation.org/UA-Profile/Security/UserToken/Server/UserNamePassword
User Token – X509 Certificate Server Facet	Security, Server	http://opcfoundation.org/UA-Profile/Security/UserToken/Server/X509Certificate
User Token – Issued Token Server Facet	Security, Server	http://opcfoundation.org/UA-Profile/Security/UserToken/Server/IssuedToken
User Token – Issued Token Windows Server Facet	Security, Server	http://opcfoundation.org/UA-Profile/Security/UserToken/Server/IssuedTokenWindows
User Token – JWT Server Facet	Security	http://opcfoundation.org/UA-Profile/Security/UserToken/Server/JsonWebToken
User Token – User Name Password Client Facet	Client, Security	http://opcfoundation.org/UA-Profile/Security/UserToken/Client/UserNamePassword
User Token – X509 Certificate Client Facet	Client, Security	http://opcfoundation.org/UA-Profile/Security/UserToken/Client/X509Certificate
User Token – Issued Token Client Facet	Client, Security	http://opcfoundation.org/UA-Profile/Security/UserToken/Client/IssuedToken
User Token – Issued Token Windows Client Facet	Client, Security	http://opcfoundation.org/UA-Profile/Security/UserToken/Client/IssuedTokenWindows
User Token – JWT Client Facet	Security	http://opcfoundation.org/UA-Profile/Security/UserToken/Client/JsonWebToken
Global Discovery Server 2017 Profile	Global Directory Service, Server	http://opcfoundation.org/UA-Profile/Server/GlobalDiscovery2017
Global Discovery and Certificate Mgmt 2017 Server	Global Directory Service, Server	http://opcfoundation.org/UA-Profile/Server/GlobalDiscoveryAndCertificateManagement2017
Global Certificate Management Client 2017 Profile	Client, Global Directory Service	http://opcfoundation.org/UA-Profile/Client/GlobalCertificateManagement2017
Global Service Authorization Request Server Facet	Global Directory Service	http://opcfoundation.org/UA-Profile/Server/GlobalServiceAuthorization
Global Service KeyCredential Pull Facet	Global Directory Service	http://opcfoundation.org/UA-Profile/Server/GlobalServiceKeyCredentials
Global Service KeyCredential Push Facet	Global Directory Service	http://opcfoundation.org/UA-Profile/Client/GlobalServiceKeyCredentials

The contents of each of the listed *Profiles* will be described in a subclause of 6.6 and where the title is equal to the *Profile* name. Each table may contain references to additional *Profiles* and or *ConformanceUnits*. If a *Profile* is referenced it means that it is completely included. The *ConformanceUnits* are referenced using their name and conformance group. For the details of the *ConformanceUnit* the reader should examine the *ConformanceUnit* details in clause 5.

6.3 Conventions for Profile definitions

Profiles have the following naming conventions:

- *Profiles* intended for OPC UA *Servers* contain the term *Server* in their titles,
- *Profiles* intended for OPC UA *Clients* contain the term *Client* in their titles
- The term Facet in the title of a *Profile* indicates that this *Profile* is expected to be part of another larger *Profile* or concerns a specific aspect of OPC UA. *Profiles* with the term Facet in their title are expected to be combined with other *Profiles* to define the complete functionality of an OPC UA *Server* or *Client*.

6.4 Profile versioning

Versioning of *Profile* is accomplished with a naming convention. Whenever a profile is revised, the year of the new revision is added to the name. Example:

Version 1	Core *Server* Facet
Version 2	Core 2017 *Server* Facet

6.5 Applications

A vendor that is developing a UA application, whether it is a *Server* application or a *Client* application, shall review the list of available *Profiles*. From this list the vendor shall select the *Profiles* that include the functionality required by the application. Typically this will be multiple *Profiles*. Conformance to a single *Profile* may not yield a complete application. In most cases multiple *Profiles* are needed to yield a useful application. All *Servers* and *Clients* shall support at least a core *Profile* (Core *Server Facet* or Core *Client Facet*) and at least one Transport *Profile*

For example an HMI *Client* application may choose to support the "Core *Client Facet*", the "UA-TCP UA-SC UA-Binary" *Profile*, the "Data Access *Client Facet*", the "DataChange Subscriber *Client* Facet" and the "*Attribute* Write *Client Facet*". If the *Client* is to be *TestLab* tested then it would also support "Base *Client* Behaviour" *Profile*. This list of *Profiles* would allow the *Client* to communicate with an OPC UA *Server* using UA-TCP/UA Security/UA binary. It would be able to subscribe for data, write to data and would support the DA data model. It would also follow the best practice guideline for behaviour. Figure 2 illustrates the *Profile* hierarchy that this application may contain. This figure is only an illustration and the represented *Profiles* may change.

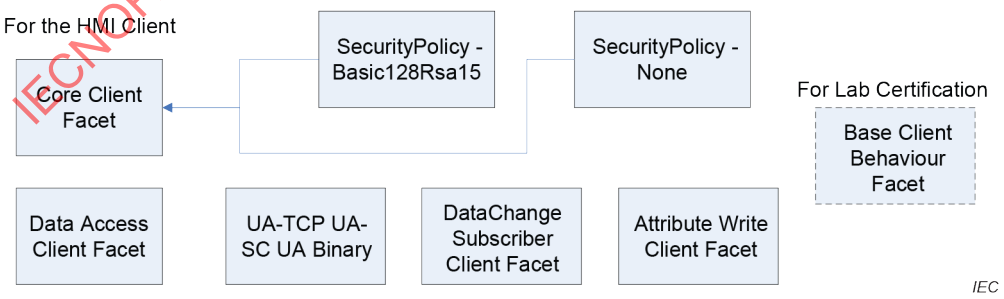


Figure 2 – HMI Client sample

All *Clients* should take into account the types of *Servers* and *Server Profiles* that they are targeted to support. Some *Servers* might not support *Subscriptions* and *Clients* should be able to fall back to Read *Services*.

A special case is a generic *Client* that is designed to communicate with a large number of *Servers* and therefore able to perform a broad range of functionality. "Standard UA *Client Profile"* has been defined for this kind of *Clients*.

Many *Clients*, however, will be specialized and do not need all of the functionality in the "Standard UA *Client Profile"* and thus would only support the limited set of functionality they require. A trend *Client*, for example, would only need functionality to subscribe to or read data.

Another example is an embedded device OPC UA *Server* application that may choose to support "Embedded UA *Server*" *Profile* and the "DataAccess *Server* Facet" *Profile*. This device would be a resource constrained device that would support UA-TCP, UA-Security, UA Binary encoding, data subscriptions and the DA data model. It may not support the optional attribute write. Figure 3 illustrates the hierarchy that this application may contain. This figure is just an illustration and the represented *Profiles* may change.

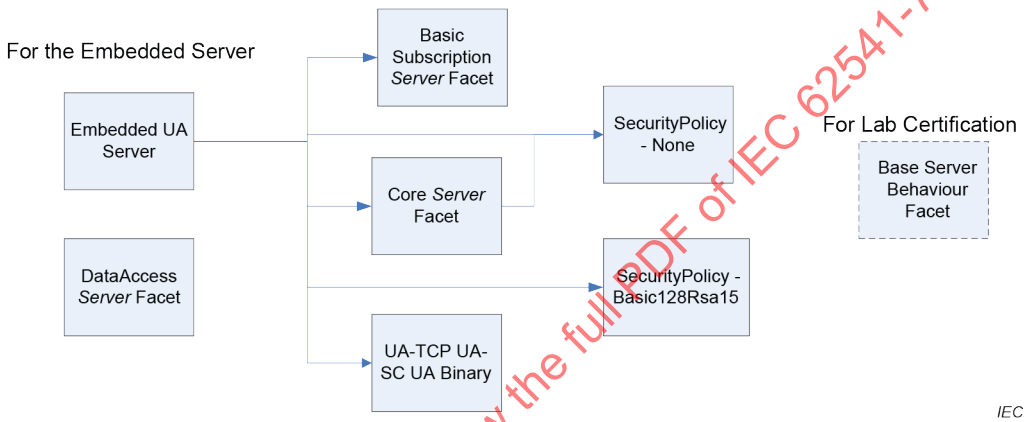


Figure 3 – Embedded Server sample

Another simple system *Server* application may choose to support: "Standard UA *Server*" *Profile* and the "DataAccess *Server* Facet" *Profile*. If the *Server* is to be lab tested then it would also support "Base *Server* Behaviour" *Profile*. This device would be a mid-level OPC UA *Server* that would support all that the embedded *Server* in the previous example supported and it would add support for an enhance level of the subscription service and support for writes. Figure 4 illustrates the hierarchy that this application may contain. This figure is just an illustration and the represented *Profile* may change.

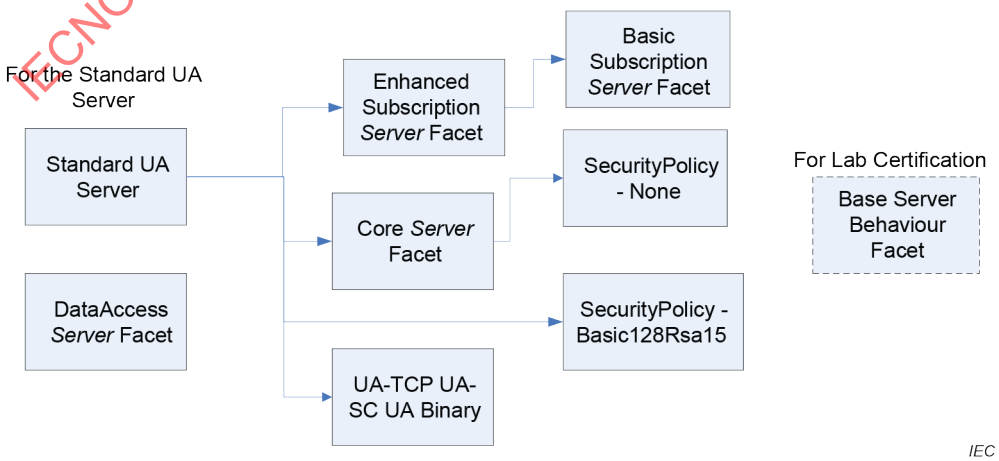


Figure 4 – Standard UA Server sample

If the example HMI *Client* were to connect to either of the example *Servers*, it may have to adjust its behavior based on the *Profile* reported by the respective *Servers*. If the HMI *Client* were communicating with the embedded device it would not be able to perform any write operations. It may also have to limit the number of subscriptions or sessions based on the performance limits of the *Server*. If the HMI *Client* is connected to the Standard *Server* it would be able to open additional windows, have higher limits on performance related items and it would be able to allow writes.

6.6 Profile tables

6.6.1 General

Subclauses 6.6.2 to 6.6.186 describe *Profiles* in a tabular format.

Each table contains three columns. The first column is a description of the conformance group that the *ConformanceUnit* is part of. This allows the reader to easily find the *ConformanceUnit*. This column may also state "*Profile*" in which case the listed item is not a *ConformanceUnit*, but an included *P rofile.* The second column is a brief description of the *ConformanceUnit* or included *Profile*. The last column indicates if the *ConformanceUnit* is optional or required.

6.6.2 Core Server Facet

Core Server Facet has been deprecated as it is superseded with Core 2017 Server Facet in v1.04.

6.6.3 Core 2017 Server Facet

Table 24 describes the details of the Core 2017 *Server* Facet. This Facet defines the core functionality required for any UA *Server* implementation. The core functionality includes the ability to discover endpoints, establish secure communication channels, create Sessions, browse the *AddressSpace* and read and/or write to *Attributes* of *Nodes*. The key requirements are: support for a single *Session*, support for the *Server* and *Server* Capabilities *Object*, all mandatory *Attributes* for *Nodes* in the *AddressSpace*, and authentication with UserName and Password. For broad applicability, it is recommended that *Servers* support multiple transport and security *Profiles*. This Facet supersedes the "Core *Server* Facet".

Table 24 – Core 2017 Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	SecurityPolicy – None	False
Profile	User Token – User Name Password *Server* Facet	False
Address Space Model	Address Space Atomicity	False
Address Space Model	Address Space Base	False
Address Space Model	Address Space Full Array Only	False
Attribute Services	*Attribute* Read	False
Attribute Services	*Attribute* Write Index	True
Attribute Services	*Attribute* Write Values	True
Base Information	Base Info Core Structure	False
Base Information	Base Info Estimated Return Time	True
Base Information	Base Info OptionSet	True
Base Information	Base Info Placeholder Modelling Rules	True
Base Information	Base Info Selection List	True
Base Information	Base Info *Server* Capabilities	True
Base Information	Base Info ValueAsText	True
Discovery Services	*Discovery* Find *Servers* Self	False
Discovery Services	*Discovery* Get Endpoints	False
Security	Security Administration	True
Security	Security Role *Server* Authorization	True
Session Services	*Session* Base	False
Session Services	*Session* General *Service* Behaviour	False
Session Services	*Session* Minimum 1	False
View *Services*	View Basic	False
View *Services*	View Minimum Continuation Point 01	False
View *Services*	View RegisterNodes	False
View *Services*	View TranslateBrowsePath	False

6.6.4 Sessionless Server Facet

Table 25 describes the details of the Sessionless *Server* Facet. Defines the use of Sessionless *Service* invocation in a *Server*.

Table 25 – Sessionless Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Discovery Services	*Discovery* Get Endpoints SessionLess	False
Session Services	*Session* Sessionless Invocation	False

6.6.5 Reverse Connect Server Facet

Table 26 describes the details of the Reverse Connect *Server* Facet. This Facet defines support of reverse connectivity in a *Server*. Usually, a connection is opened by the *Client* before starting the UA-specific handshake. This will fail, however, when *Servers* are behind firewalls with no open ports to connect to. In the reverse connectivity scenario, the *Server* opens the connection and starts with a ReverseHello message requesting that the *Client* establish a Secure Channel using this connection.

Table 26 – Reverse Connect Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Protocol and Encoding	Protocol Reverse Connect *Server*	False

6.6.6 Base Server Behaviour Facet

Table 27 describes the details of the Base *Server* Behaviour Facet. This Facet defines best practices for the configuration and management of *Servers* when they are deployed in a production environment. It provides the ability to enable or disable certain protocols and to configure the *Discovery Server* and specify where this *Server* shall be registered.

Table 27 – Base Server Behaviour Facet

Group	Conformance Unit / *Profile* Title	Optional
Discovery Services	*Discovery* Configuration	False
Protocol and Encoding	Protocol Configuration	False
Security	Security Administration	False
Security	Security Administration – XML Schema	False
Security	Security *Certificate* Administration	False

6.6.7 Request State Change Server Facet

Table 28 describes the details of the Request State Change *Server* Facet. This Facet specifies the support of the RequestServerStateChange *Method*.

Table 28 – Request State Change Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Base Information	Base Info RequestServerStateChange *Method*	False

6.6.8 Subnet Discovery Server Facet

Table 29 describes the details of the Subnet *Discovery Server* Facet. Support of this Facet enables discovery of the *Server* on a subnet using mDNS. This functionality is only applicable when *Servers* do not register with an LDS.

Table 29 – Subnet Discovery Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Discovery Services	*Discovery Server* Announcement using mDNS	False

6.6.9 Global Certificate Management Server Facet

Table 30 describes the details of the Global *Certificate* Management *Server* Facet. This Facet defines the capability to interact with a Global *Certificate* Management *Server* to obtain an initial or renewed *Certificate* and Trust Lists.

Table 30 – Global Certificate Management Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Security	Push Model for Global *Certificate* and TrustList Management	False

6.6.10 Authorization Service Server Facet

Table 31 describes the details of the Authorization *Service Server* Facet. This Facet defines the support for configuring the necessary information to validate access tokens when presented by a Client during session establishment. Access Tokens are issued by Authorization *Services*.

Table 31 – Authorization Service Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Security	Authorization Service Configuration Server	False

6.6.11 KeyCredential Service Server Facet

Table 32 describes the details of the KeyCredential *Service Server* Facet. This Facet defines the capability to interact with a KeyCredential *Service* to obtain KeyCredentials. For example KeyCredentials are needed to access an Authorization *Service* or a Broker. The KeyCredential *Service* is typically part of a system-wide tool, like a GDS that also manages Applications, Access Tokens, and *Certificates*.

Table 32 – KeyCredential Service Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Security	Push Model for KeyCredential *Service*	False

6.6.12 Attribute WriteMask Server Facet

Table 33 describes the details of the *Attribute* WriteMask *Server* Facet. This Facet defines the capability to update characteristics of individual *Nodes* in the *AddressSpace* by allowing writing to *Node Attributes*. It requires support for authenticating user access as well as providing information related to access rights in the *AddressSpace* and actually restricting the access rights as described.

Table 33 – Attribute WriteMask Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	Security User Access Control Base	False
Address Space Model	Address Space UserWriteMask	False
Address Space Model	Address Space UserWriteMask Multilevel	True
Address Space Model	Address Space WriteMask	False

6.6.13 File Access Server Facet

Table 34 describes the details of the File Access *Server* Facet. This Facet specifies the support of exposing File information via the defined FileType. This includes reading of file as well as optionally writing of file data.

Table 34 – File Access Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Base Information	Base Info FileType Base	False
Base Information	Base Info FileType Write	True

6.6.14 Documentation Server Facet

Table 35 describes the details of the Documentation *Server* Facet. This Facet defines a list of user documentation that a server application should provide.

Table 35 – Documentation Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Miscellaneous	Documentation – Installation	False
Miscellaneous	Documentation – Multiple Languages	True
Miscellaneous	Documentation – On-line	True
Miscellaneous	Documentation – Supported *Profiles*	True
Miscellaneous	Documentation – Trouble Shooting Guide	True
Miscellaneous	Documentation – Users Guide	False

6.6.15 Embedded DataChange Subscription Server Facet

Table 36 describes the details of the Embedded DataChange *Subscription Server* Facet. This Facet specifies the minimum level of support for data change notifications within subscriptions. It includes limits which minimize memory and processing overhead required to implement the Facet. This Facet includes functionality to create, modify and delete Subscriptions and to add, modify and remove Monitored Items. As a minimum for each *Session*, *Servers* shall support one *Subscription* with up to two items. In addition, support for two parallel Publish requests is required. This Facet is geared for a platform such as the one provided by the Micro Embedded Device *Server Profile* in which memory is limited and needs to be managed.

Table 36 – Embedded DataChange Subscription Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Monitored Item *Services*	Monitor Basic	False
Monitored Item *Services*	Monitor Items 2	False
Monitored Item *Services*	Monitor QueueSize_1	False
Monitored Item *Services*	Monitor Value Change	False
Subscription Services	*Subscription* Basic	False
Subscription Services	*Subscription* Minimum 1	False
Subscription Services	*Subscription* Publish Discard Policy	False
Subscription Services	*Subscription* Publish Min 02	False

6.6.16 Standard DataChange Subscription Server Facet

Standard DataChange Subscription Server Facet has been deprecated as it is superseded with Standard DataChange Subscription 2017 Server Facet in v1.04.

6.6.17 Standard DataChange Subscription 2017 Server Facet

Table 37 describes the details of the Standard DataChange *Subscription* 2017 *Server* Facet. This Facet specifies the standard support of subscribing to data changes and extends features and limits defined by the Embedded Data Change *Subscription* Facet. See *ConformanceUnits* for these limits. Note that the *Method* Call *Service* is only required for the *Methods* defined in this Facet. This Facet supersedes the "Standard DataChange *Subscription Server* Facet".

Table 37 – Standard DataChange Subscription 2017 Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	Embedded DataChange *Subscription Server* Facet	False
Base Information	Base Info GetMonitoredItems *Method*	False
Base Information	Base Info ResendData *Method*	False
Method Services	*Method* Call	False
Monitored Item *Services*	Monitor Items 10	False
Monitored Item *Services*	Monitor Items 100	False
Monitored Item *Services*	Monitor MinQueueSize_02	False
Monitored Item *Services*	Monitor Triggering	False
Monitored Item *Services*	Monitored Items Deadband Filter	False
Subscription Services	*Subscription* Minimum 02	False
Subscription Services	*Subscription* Publish Min 05	False

6.6.18 Enhanced DataChange Subscription Server Facet

Enhanced DataChange *Subscription Server* Facet has been deprecated as it is superseded with Enhanced DataChange Subscription 2017 Server Facet in v1.04.

6.6.19 Enhanced DataChange Subscription 2017 Server Facet

Table 38 describes the details of the Enhanced DataChange *Subscription* 2017 *Server* Facet. This Facet specifies an enhanced support of subscribing to data changes. It is part of the Standard UA *Server* 2017 *Profile*. This Facet increases the limits defined by the Standard Data Change *Subscription* 2017 *Server* Facet.

Table 38 – Enhanced DataChange Subscription 2017 Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	Standard DataChange *Subscription* 2017 *Server* Facet	False
Monitored Item *Services*	Monitor Items 500	False
Monitored Item *Services*	Monitor MinQueueSize_05	False
Subscription Services	*Subscription* Minimum 05	False
Subscription Services	*Subscription* Publish Min 10	False

6.6.20 Durable Subscription Server Facet

Table 39 describes the details of the Durable *Subscription Server* Facet. This Facet specifies support of durable storage of data and events even when *Clients* are disconnected. This Facet implies support of any of the DataChange or *Event Subscription* Facets.

Table 39 – Durable Subscription Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Subscription Services	*Subscription* Durable	False
Subscription Services	*Subscription* Durable StorageLevel High	True
Subscription Services	*Subscription* Durable StorageLevel Medium	True
Subscription Services	*Subscription* Durable StorageLevel Small	True

6.6.21 Data Access Server Facet

Table 40 describes the details of the Data Access *Server* Facet. This Facet specifies the support for an Information Model used to provide industrial automation data. This model defines standard structures for analog and discrete data items and their quality of service. This Facet extends the Core *Server* Facet which includes support of the basic *AddressSpace* behaviour.

Table 40 – Data Access Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Data Access	Data Access AnalogItems	True
Data Access	Data Access ArrayItemType	True
Data Access	Data Access Complex Number	True
Data Access	Data Access DataItems	False
Data Access	Data Access DoubleComplex Number	True
Data Access	Data Access MultiState	True
Data Access	Data Access MultiStateValueDiscrete	True
Data Access	Data Access PercentDeadband	True
Data Access	Data Access Semantic Changes	True
Data Access	Data Access TwoState	True

6.6.22 ComplexType Server Facet

ComplexType Server Facet has been deprecated as it is superseded with ComplexType 2017 Server Facet in v1.04.

6.6.23 ComplexType 2017 Server Facet

Table 41 describes the details of the ComplexType 2017 *Server* Facet. This Facet extends the Core *Server* Facet to include *Variables* with structured data, i.e. data that are composed of multiple elements such as a structure and where the individual elements are exposed as component variables. Support of this Facet requires the implementation of structured DataTypes and *Variables* that make use of these DataTypes. The Read, Write and Subscriptions service set shall support the encoding and decoding of these structured DataTypes. As an option the *Server* can also support alternate encodings, such as an XML encoding when the binary protocol is currently used and vice-versa.

Table 41 – ComplexType 2017 Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Address Space Model	Address Space DataTypeDefinition *Attribute*	False
Attribute Services	*Attribute* Alternate Encoding	True
Attribute Services	*Attribute* Read Complex	False
Attribute Services	*Attribute* Write Complex	False
Monitored Item *Services*	Monitor Alternate Encoding	True
Monitored Item Services	Monitor Complex Value	True

6.6.24 Standard Event Subscription Server Facet

Table 42 describes the details of the Standard *Event Subscription Server* Facet. This Facet specifies the standard support for subscribing to events and is intended to supplement any of the FullFeatured *Profiles*. Support of this Facet requires the implementation of *Event* Types representing the Events that the *Server* can report and their specific fields. It also requires at least the *Server Object* to have the *EventNotifier Attribute* set. It includes the *Services* to Create, Modify and Delete Subscriptions and to Add, Modify and Remove Monitored Items for *Object Nodes* with an "*EventNotifier Attribute*". Creating a monitoring item may include a filter that includes SimpleAttribute FilterOperands and a select list of Operators. The operators include: Equals, IsNull, GreaterThan, LessThan, GreaterThanOrEqual, LessThanOrEqual, Like, Not, Between, InList, And, Or, Cast, BitwiseAnd, BitwiseOr and TypeOf. Support of more complex filters is optional. This Facet has been updated to include several optional Base Information *ConformanceUnits*. These *ConformanceUnits* are optional to allow for backward compatibility, in the future these optional *ConformanceUnits* will become required, and so it is highly recommended that all servers support them.

Table 42 – Standard Event Subscription Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Address Space Model	Address Space Events	False
Base Information	Base Info Device Failure	True
Base Information	Base Info EventQueueOverflow EventType	True
Base Information	Base Info Progress Events	True
Base Information	Base Info SemanticChange	True
Base Information	Base Info System Status	True
Base Information	Base Info System Status Underlying System	True
Monitored Item *Services*	Monitor Basic	False
Monitored Item *Services*	Monitor Complex *Event* Filter	True
Monitored Item *Services*	Monitor Events	False
Monitored Item *Services*	Monitor Items 10	False
Monitored Item *Services*	Monitor QueueSize_ServerMax	False
Subscription Services	*Subscription* Basic	False
Subscription Services	*Subscription* Minimum 02	False
Subscription Services	*Subscription* Publish Discard Policy	False
Subscription Services	*Subscription* Publish Min 05	False

6.6.25 Address Space Notifier Server Facet

Table 43 describes the details of the Address Space Notifier *Server* Facet. This Facet requires the support of a hierarchy of *Object Nodes* that are notifiers and *Nodes* that are event sources. The hierarchy is commonly used as a way to organize a plant into areas that can be managed by different operators.

Table 43 – Address Space Notifier Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Address Space Model	Address Space Notifier Hierarchy	False
Address Space Model	Address Space Source Hierarchy	False

6.6.26 A & C Base Condition Server Facet

Table 44 describes the details of the A & C Base *Condition Server* Facet. This Facet requires basic support for *Conditions*. Information about *Conditions* is provided through *Event* notifications and thus this Facet builds upon the Standard *Event Subscription Server* Facet. *Conditions* that are in an "interesting" state (as defined by the *Server*) can be refreshed using the Refresh *Method*, which requires support for the *Method Server* Facet. Optionally the server may also provide support for *Condition* classes

Table 44 – A & C Base Condition Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	*Method Server* Facet	False
Profile	Standard *Event Subscription Server* Facet	False
Alarms and *Conditions*	A & C Basic	False
Alarms and *Conditions*	A & C *Condition* Sub-Classes	True
Alarms and *Conditions*	A & C ConditionClasses	True
Alarms and *Conditions*	A & C Refresh	False

6.6.27 A & C Refresh2 Server Facet

Table 45 describes the details of the A & C Refresh2 *Server* Facet. This Facet enhances the A & C Base *Condition Server* Facet with support of the ConditionRefresh2 *Method*.

Table 45 – A & C Refresh2 Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	A & C Base *Condition Server* Facet	False
Alarms and *Conditions*	A & C Refresh2	False

6.6.28 A & C Address Space Instance Server Facet

Table 46 describes the details of the A & C Address Space Instance *Server* Facet. This Facet specifies the support required for a *Server* to expose *Alarms* and *Conditions* in its *AddressSpace*. This includes the A & C *AddressSpace* information model.

Table 46 – A & C Address Space Instance Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Alarms and *Conditions*	A & C Instances	False

6.6.29 A & C Enable Server Facet

Table 47 describes the details of the A & C Enable *Server* Facet. This Facet requires the enabling and disabling of *Conditions*. This Facet builds upon the A&C Base *Condition Server* Facet. Enabling and disabling also requires that instances of these ConditionTypes exist in the *AddressSpace* since the enable *Method* can only be invoked on an instance of the *Condition*

Table 47 – A & C Enable Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	A & C Base *Condition Server* Facet	False
Alarms and *Conditions*	A & C Enable	False
Alarms and *Conditions*	A & C Instances	False

6.6.30 A & C AlarmMetrics Server Facet

Table 48 describes the details of the A & C AlarmMetrics *Server* Facet. This Facet requires support for AlarmMetrics. AlarmMetrics expose status and potential issues in the alarm system. A *Server* can provide these metrics at various levels (operator station, plant area, overall system etc.).

Table 48 – A & C AlarmMetrics Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Alarms and *Conditions*	A & C *Alarm* Metrics	False

6.6.31 A & C Alarm Server Facet

Table 49 describes the details of the A & C *Alarm Server* Facet. This Facet requires support for *Alarms*. *Alarms* extend the ConditionType by adding an Active state which indicates when something in the system requires attention by an Operator. This Facet builds upon the A&C Base *Condition Server* Facet. This facet requires that discrete AlarmTypes be supported, it also allows for optional support of shelving, alarm comments and other discrete AlarmTypes such as Trip or Off-Normal.

Table 49 – A & C Alarm Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	A & C Base *Condition Server* Facet	False
Alarms and *Conditions*	A & C *Alarm*	False
Alarms and *Conditions*	A & C Audible Sound	True
Alarms and *Conditions*	A & C Comment	True
Alarms and *Conditions*	A & C Discrepancy	True
Alarms and *Conditions*	A & C Discrete	False
Alarms and *Conditions*	A & C First in Group *Alarm*	True
Alarms and *Conditions*	A & C OffNormal	True
Alarms and *Conditions*	A & C On-Off Delay	True
Alarms and *Conditions*	A & C Out Of Service	True
Alarms and *Conditions*	A & C Re-Alarming	True
Alarms and *Conditions*	A & C Shelving	True
Alarms and *Conditions*	A & C Silencing	True
Alarms and *Conditions*	A & C Suppression	True
Alarms and *Conditions*	A & C Suppression by Operator	True
Alarms and *Conditions*	A & C SystemOffNormal	True
Alarms and *Conditions*	A & C Trip	True

6.6.32 A & C Acknowledgeable Alarm Server Facet

Table 50 describes the details of the A & C Acknowledgeable *Alarm Server* Facet. This Facet requires support for Acknowledgement of active *Alarms*. This Facet builds upon the A & C *Alarm Server* Facet. Acknowledgement requires support of the Acknowledge *Method* and the Acknowledged state. Support of the Confirmed state and the Confirm *Method* is optional.

Table 50 – A & C Acknowledgeable Alarm Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	A & C *Alarm Server* Facet	False
Alarms and *Conditions*	A & C Acknowledge	False
Alarms and *Conditions*	A & C Confirm	True

6.6.33 A & C Exclusive Alarming Server Facet

Table 51 describes the details of the A & C Exclusive Alarming *Server* Facet. This Facet requires support for *Alarms* with multiple sub-states that identify different limit *Conditions*. This facet builds upon the A&C *Alarm Server* Facet. The term exclusive means only one sub-state can be active at a time. For example, a temperature exceeds the HighHigh limit the associated exclusive LevelAlarm will be in the HighHigh sub-state and not in the High sub-state. This Facet requires that a *Server* support at least one of the optional *Alarm* models: Limit, RateOfChange or Deviation.

Table 51 – A & C Exclusive Alarming Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	A & C *Alarm Server* Facet	False
Alarms and *Conditions*	A & C Exclusive Deviation	True
Alarms and *Conditions*	A & C Exclusive Level	True
Alarms and *Conditions*	A & C Exclusive Limit	False
Alarms and *Conditions*	A & C Exclusive RateOfChange	True

6.6.34 A & C Non-Exclusive Alarming Server Facet

Table 52 describes the details of the A & C Non-Exclusive Alarming *Server* Facet. This Facet requires support for *Alarms* with multiple sub-states that identify different limit *Conditions*. This Facet builds upon the A&C *Alarm Server* Facet. The term non-exclusive means more than one sub-state can be active at a time. For example, if a temperature exceeds the HighHigh limit the associated non-exclusive LevelAlarm will be in both the High and the HighHigh sub-state. This Facet requires that a server support at least one of the optional alarm models: Limit, RateOfChange or Deviation.

Table 52 – A & C Non-Exclusive Alarming Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	A & C *Alarm Server* Facet	False
Alarms and *Conditions*	A & C Non-Exclusive Deviation	True
Alarms and *Conditions*	A & C Non-Exclusive Level	True
Alarms and *Conditions*	A & C Non-Exclusive Limit	False
Alarms and *Conditions*	A & C Non-Exclusive RateOfChange	True

6.6.35 A & C Previous Instances Server Facet

Table 53 describes the details of the A & C Previous Instances *Server* Facet. This Facet requires support for *Conditions* with previous states that still require action on the part of the operator. This Facet builds upon the A&C Base *Condition Server* Facet. A common use case for this Facet is a safety critical system that requires that all *Alarms* be acknowledged even if it the original problem goes away and the *Alarm* returns to the inactive state. In these cases, the previous state with active *Alarm* is still reported by the *Server* until the Operator acknowledges it. When a *Condition* has previous states it will produce events with different Branch identifiers. When previous state no longer needs attention the branch will disappear.

Table 53 – A & C Previous Instances Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	A & C Base *Condition Server* Facet	False
Alarms and *Conditions*	A & C Branch	False

6.6.36 A & C Dialog Server Facet

Table 54 describes the details of the A & C Dialog *Server* Facet. This Facet requires support of Dialog *Conditions*. This Facet builds upon the A & C BaseCondition *Server* Facet Dialogs are ConditionTypes used to request user input. They are typically used when a *Server* has entered some state that requires intervention by a *Client*. For example, a *Server* monitoring a paper machine indicates that a roll of paper has been wound and is ready for inspection. The *Server* would activate a Dialog *Condition* indicating to the user that an inspection is required.

Once the inspection has taken place the user responds by informing the *Server* of an accepted or unaccepted inspection allowing the process to continue.

Table 54 – A & C Dialog Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	A & C Base *Condition Server* Facet	False
Alarms and *Conditions*	A & C Dialog	False

6.6.37 A & C CertificateExpiration Server Facet

Table 55 describes the details of the A & C CertificateExpiration *Server* Facet. This Facet requires support of the CertificateExpirationAlarmType. It is used to inform *Clients* when the *Server*'s *Certificate* is within the defined expiration period.

Table 55 – A & C CertificateExpiration Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	A & C Base *Condition Server* Facet	False
Alarms and *Conditions*	A & C Acknowledge	False
Alarms and *Conditions*	A & C *Alarm*	False
Alarms and *Conditions*	A & C CertificateExpiration	False
Alarms and *Conditions*	A & C Comment	True
Alarms and *Conditions*	A & C Confirm	True
Alarms and *Conditions*	A & C Shelving	True

6.6.38 A & E Wrapper Facet

Table 56 describes the details of the A & E Wrapper Facet. This Facet specifies the requirements for a UA *Server* that wraps an OPC *Alarm* & *Event* (AE) *Server* (COM). This *Profile* identifies the sub-set of the UA *Alarm* & *Condition* model which is provided by the COM OPC AE specification. It is intended to provide guidance to developers who are creating servers that front end existing applications. It is important to note that some OPC A&E COM *Servers* may not support all of the functionality provided by an OPC UA A&C server, in these cases similar functionality maybe available via some non-OPC interface. For example if an A&E COM server does not support sending *Alarm* Acknowledgement messages to the system that it is obtaining alarm information from, this functionality may be available via some out of scope features in the underlying *Alarm* system. Another possibility is that the underlying system does not require acknowledgements or automatically acknowledges the alarm.

Table 56 – A & E Wrapper Facet

Group	Conformance Unit / *Profile* Title	Optional
Address Space Model	Address Space Events	False
Address Space Model	Address Space Notifier Hierarchy	False
Address Space Model	Address Space Source Hierarchy	False
Alarms and *Conditions*	A & C Acknowledge	False
Alarms and *Conditions*	A & C *Alarm*	False
Alarms and *Conditions*	A & C Basic	False
Alarms and *Conditions*	A & C ConditionClasses	False
Alarms and *Conditions*	A & C Refresh	False
Alarms and *Conditions*	A & E Wrapper Mapping	False
Monitored Item *Services*	Monitor Basic	False
Monitored Item *Services*	Monitor Complex *Event* Filter	False
Monitored Item *Services*	Monitor Events	False
Monitored Item *Services*	Monitor Items 2	False
Monitored Item *Services*	Monitor QueueSize_ServerMax	False
Subscription Services	*Subscription* Basic	False
Subscription Services	*Subscription* Minimum 1	False
Subscription Services	*Subscription* Publish Discard Policy	False
Subscription Services	*Subscription* Publish Min 02	False

6.6.39 Method Server Facet

Table 57 describes the details of the *Method Server* Facet. This Facet specifies the support of *Method* invocation via the Call service. Methods are "lightweight" functions which are similar to the methods of a class found in any object-oriented programming language. A *Method* can have its scope bounded by an owning *Object* or an owning *ObjectType*. Methods with an *ObjectType* as their scope are similar to static methods in a class.

Table 57 – Method Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Address Space Model	Address Space *Method*	False
Method Services	*Method* Call	False

6.6.40 Auditing Server Facet

Table 58 describes the details of the Auditing *Server* Facet. This Facet requires the support of Auditing which includes the Standard *Event Subscription Server* Facet. Support of this Facet requires that Audit Events be produced when a client performs some action to change the state of the server, such as changing the *AddressSpace*, inserting or updating a value etc. The auditEntryId passed by the *Client* is a field contained in every Audit *Event* and allows actions to be traced across multiple systems. The Audit *Event* Types and their fields shall be exposed in the *Server*'s *AddressSpace*

Table 58 – Auditing Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	Standard *Event Subscription Server* Facet	False
Auditing	Auditing Base	False

6.6.41 Node Management Server Facet

Table 59 describes the details of the *Node* Management *Server* Facet. This Facet requires the support of the *Services* that allow the *Client* to add, modify and delete *Nodes* in the *AddressSpace*. These *Services* provide an interface which can be used to configure *Servers*. This means all changes to the *AddressSpace* are expected to persist even after the *Client* has disconnected from the *Server*

Table 59 – Node Management Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Address Space Model	Address Space Base	False
Base Information	Base Info Model Change	False
Base Information	Base Info Type System	False
Node Management *Services*	*Node* Management Add *Node*	False
Node Management *Services*	*Node* Management Add Ref	False
Node Management *Services*	*Node* Management Delete *Node*	False
Node Management *Services*	*Node* Management Delete Ref	False

6.6.42 User Role Base Server Facet

Table 60 describes the details of the User Role Base *Server* Facet. This Facet defines support of the OPC UA Information Model to expose configured user roles and permissions.

Table 60 – User Role Base Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Security	Security Role *Server* Base	False

6.6.43 User Role Management Server Facet

Table 61 describes the details of the User Role Management *Server* Facet. This Facet defines support of the OPC UA approach to manage user roles and permissions and to grant access to *Nodes* and *Services* based on the assigned roles and permissions.

Table 61 – User Role Management Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	User Role Base *Server* Facet	False
Security	Security Role *Server* DefaultRolePermissions	False
Security	Security Role *Server* IdentityManagement	False
Security	Security Role *Server* Management	False
Security	Security Role *Server* Restrict Applications	True
Security	Security Role *Server* Restrict Endpoints	True
Security	Security Role *Server* RolePermissions	True
Security	Security Role Well Known	False

6.6.44 State Machine Server Facet

Table 62 describes the details of the State Machine *Server* Facet. This Facet defines support of StateMachines based on the types in IEC 62541-5.

Table 62 – State Machine Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Base Information	Base Info Available States and Transitions	True
Base Information	Base Info Finite State Machine Instance	True
Base Information	Base Info State Machine Instance	False

6.6.45 Client Redundancy Server Facet

Table 63 describes the details of the *Client* Redundancy *Server* Facet. This Facet defines the *Server* actions that are required for support of redundant *Clients*. Support of this Facet requires the implementation of the TransferSubscriptions *Service* which allows the transfer of Subscriptions from one *Client*'s *Session* to another *Client*'s *Session*.

Table 63 – Client Redundancy Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Subscription Services	*Subscription* Transfer	False

6.6.46 Redundancy Transparent Server Facet

Table 64 describes the details of the Redundancy Transparent *Server* Facet. This Facet requires support for transparent redundancy. If *Servers* implement transparent redundancy then the failover from one *Server* to another is transparent to the *Client* such that the *Client* is unaware that a failover has occurred; the *Client* does not need to do anything at all to keep data flowing. This type of redundancy is usually a hardware solution.

Table 64 – Redundancy Transparent Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Redundancy	Redundancy *Server* Transparent	False

6.6.47 Redundancy Visible Server Facet

Table 65 describes the details of the Redundancy Visible *Server* Facet. This Facet specifies the support for non-transparent redundancy. Failover for this type of redundancy requires the *Client* to monitor *Server* status and to switch to a backup *Server* if it detects a failure. The *Server* shall expose the methods of failover it supports (cold, warm or hot). The failover method tells the *Client* what it shall do when connecting to a *Server* and when a failure occurs. Cold redundancy requires a *Client* to reconnect to a backup *Server* after the initial *Server* has failed. Warm redundancy allows a *Client* to connect to multiple *Servers*, but only one *Server* will be providing values. In hot redundancy multiple *Servers* are able to provide data and a *Client* can connect to multiple *Servers* for the data.

Table 65 – Redundancy Visible Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Redundancy	Redundancy *Server*	False

6.6.48 Historical Raw Data Server Facet

Table 66 describes the details of the Historical Raw Data *Server* Facet. This Facet defines the basic functionality when supporting historical data access for raw data.

Table 66 – Historical Raw Data Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Attribute Services	*Attribute* Historical Read	False
Historical Access	Historical Access Data Max *Nodes* Read Continuation Point	False
Historical Access	Historical Access Read Raw	False
Historical Access	Historical Access ServerTimestamp	True

6.6.49 Historical Aggregate Server Facet

Table 67 describes the details of the Historical Aggregate *Server* Facet. This Facet indicates that the server supports aggregate processing to produce derived values from raw historical data.

Table 67 – Historical Aggregate Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Aggregates	Aggregate – AnnotationCount	True
Aggregates	Aggregate – Average	True
Aggregates	Aggregate – Count	True
Aggregates	Aggregate – Custom	True
Aggregates	Aggregate – Delta	True
Aggregates	Aggregate – DeltaBounds	True
Aggregates	Aggregate – DurationBad	True
Aggregates	Aggregate – DurationGood	True
Aggregates	Aggregate – DurationInStateNonZero	True
Aggregates	Aggregate – DurationInStateZero	True
Aggregates	Aggregate – End	True
Aggregates	Aggregate – EndBound	True
Aggregates	Aggregate – Interpolative	True
Aggregates	Aggregate – Maximum	True
Aggregates	Aggregate – Maximum2	True
Aggregates	Aggregate – MaximumActualTime	True
Aggregates	Aggregate – MaximumActualTime2	True
Aggregates	Aggregate – Minimum	True
Aggregates	Aggregate – Minimum2	True
Aggregates	Aggregate – MinimumActualTime	True
Aggregates	Aggregate – MinimumActualTime2	True
Aggregates	Aggregate – NumberOfTransitions	True
Aggregates	Aggregate – PercentBad	True
Aggregates	Aggregate – PercentGood	True
Aggregates	Aggregate – Range	True
Aggregates	Aggregate – Range2	True
Aggregates	Aggregate – StandardDeviationPopulation	True
Aggregates	Aggregate – StandardDeviationSample	True
Aggregates	Aggregate – Start	True
Aggregates	Aggregate – StartBound	True
Aggregates	Aggregate – TimeAverage	True
Aggregates	Aggregate – TimeAverage2	True
Aggregates	Aggregate – Total	True
Aggregates	Aggregate – Total2	True
Aggregates	Aggregate – VariancePopulation	True
Aggregates	Aggregate – VarianceSample	True
Aggregates	Aggregate – WorstQuality	True
Aggregates	Aggregate – WorstQuality2	True
Aggregates	Aggregate Historical Configuration	True
Aggregates	Aggregate Master Configuration	False
Attribute Services	*Attribute* Historical Read	False
Historical Access	Historical Access Aggregates	False
Historical Access	Historical Access Data Max *Nodes* Read Continuation Point	False

6.6.50 Historical Data AtTime Server Facet

Table 68 describes the details of the Historical Data AtTime *Server* Facet. This Facet indicates that the historical *Server* supports reading data by specifying specific timestamps.

Table 68 – Historical Data AtTime Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Attribute Services	*Attribute* Historical Read	False
Historical Access	Historical Access Data Max *Nodes* Read Continuation Point	False
Historical Access	Historical Access Time Instance	False

6.6.51 Historical Access Modified Data Server Facet

Table 69 describes the details of the Historical Access Modified Data *Server* Facet. This Facet defines support of reading modified historical values (values that where modified or inserted).

Table 69 – Historical Access Modified Data Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Attribute Services	*Attribute* Historical Read	False
Historical Access	Historical Access Modified Values	False

6.6.52 Historical Annotation Server Facet

Table 70 describes the details of the Historical Annotation *Server* Facet. This Facet defines support for the storage and retrieval of annotations for historical data.

Table 70 – Historical Annotation Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Attribute Services	*Attribute* Historical Read	False
Attribute Services	*Attribute* Historical Update	False
Historical Access	Historical Access Annotations	False

6.6.53 Historical Data Insert Server Facet

Table 71 describes the details of the Historical Data Insert *Server* Facet. This Facet includes Historical Data Insert functionality.

Table 71 – Historical Data Insert Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Attribute Services	*Attribute* Historical Update	False
Historical Access	Historical Access Insert Value	False
Historical Access	Historical Access ServerTimestamp	True

6.6.54 Historical Data Update Server Facet

Table 72 describes the details of the Historical Data Update *Server* Facet. This Facet includes Historical Data Update functionality.

Table 72 – Historical Data Update Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Attribute Services	*Attribute* Historical Update	False
Historical Access	Historical Access ServerTimestamp	True
Historical Access	Historical Access Update Value	False

6.6.55 Historical Data Replace Server Facet

Table 73 describes the details of the Historical Data Replace *Server* Facet. This Facet includes Historical Data Replace functionality.

Table 73 – Historical Data Replace Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Attribute Services	*Attribute* Historical Update	False
Historical Access	Historical Access Replace Value	False
Historical Access	Historical Access ServerTimestamp	True

6.6.56 Historical Data Delete Server Facet

Table 74 describes the details of the Historical Data Delete *Server* Facet. This Facet includes Historical Data Delete functionality.

Table 74 – Historical Data Delete Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Attribute Services	*Attribute* Historical Update	False
Historical Access	Historical Access Delete Value	False

6.6.57 Historical Access Structured Data Server Facet

Table 75 describes the details of the Historical Access Structured Data *Server* Facet. This Facet indicates that the *Server* supports storage and retrieval of structured values for all supported access types. If a listed access type is supported then the corresponding optional *ConformanceUnit* shall be supported.

Table 75 – Historical Access Structured Data Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Historical Access	Historical Access Structured Data Delete	True
Historical Access	Historical Access Structured Data Insert	True
Historical Access	Historical Access Structured Data Read Modified	True
Historical Access	Historical Access Structured Data Read Raw	False
Historical Access	Historical Access Structured Data Replace	True
Historical Access	Historical Access Structured Data Time Instance	True
Historical Access	Historical Access Structured Data Update	True

6.6.58 Base Historical Event Server Facet

Table 76 describes the details of the Base Historical *Event Server* Facet. This Facet defines the server requirements to support basic Historical *Event* functionality, including simple filtering and general access.

Table 76 – Base Historical Event Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Attribute Services	*Attribute* Historical Read	False
Historical Access	Historical Access *Event* Max Events Read Continuation Point	False
Historical Access	Historical Access Events	False

6.6.59 Historical Event Update Server Facet

Table 77 describes the details of the Historical *Event* Update *Server* Facet. This Facet includes Historical *Event* update access functionality.

Table 77 – Historical Event Update Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Attribute Services	*Attribute* Historical Update	False
Historical Access	Historical Access Update *Event*	False

6.6.60 Historical Event Replace Server Facet

Table 78 describes the details of the Historical *Event* Replace *Server* Facet. This Facet includes Historical *Event* replace access functionality.

Table 78 – Historical Event Replace Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Attribute Services	*Attribute* Historical Update	False
Historical Access	Historical Access Replace *Event*	False

6.6.61 Historical Event Insert Server Facet

Table 79 describes the details of the Historical *Event* Insert *Server* Facet. This Facet includes Historical *Event* insert access functionality.

Table 79 – Historical Event Insert Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Attribute Services	*Attribute* Historical Update	False
Historical Access	Historical Access Insert *Event*	False

6.6.62 Historical Event Delete Server Facet

Table 80 describes the details of the Historical *Event* Delete *Server* Facet. This Facet includes Historical *Event* delete access functionality

Table 80 – Historical Event Delete Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Attribute Services	*Attribute* Historical Update	False
Historical Access	Historical Access Delete *Event*	False

6.6.63 Aggregate Subscription Server Facet

Table 81 describes the details of the Aggregate *Subscription Server* Facet. This Facet defines the handling of the aggregate filter when subscribing for *Attribute* values.

Table 81 – Aggregate Subscription Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	Standard DataChange *Subscription Server* Facet	False
Aggregates	Aggregate *Subscription* – AnnotationCount	True
Aggregates	Aggregate *Subscription* – Average	True
Aggregates	Aggregate *Subscription* – Count	True
Aggregates	Aggregate *Subscription* – Custom	True
Aggregates	Aggregate *Subscription* – Delta	True
Aggregates	Aggregate *Subscription* – DeltaBounds	True
Aggregates	Aggregate *Subscription* – DurationBad	True
Aggregates	Aggregate *Subscription* – DurationGood	True
Aggregates	Aggregate *Subscription* – DurationInStateNonZero	True
Aggregates	Aggregate *Subscription* – DurationInStateZero	True
Aggregates	Aggregate *Subscription* – End	True
Aggregates	Aggregate *Subscription* – EndBound	True
Aggregates	Aggregate *Subscription* – Filter	False
Aggregates	Aggregate *Subscription* – Interpolative	True
Aggregates	Aggregate *Subscription* – Maximum	True
Aggregates	Aggregate *Subscription* – Maximum2	True
Aggregates	Aggregate *Subscription* – MaximumActualTime	True
Aggregates	Aggregate *Subscription* – MaximumActualTime2	True
Aggregates	Aggregate *Subscription* – Minimum	True
Aggregates	Aggregate *Subscription* – Minimum2	True
Aggregates	Aggregate *Subscription* – MinimumActualTime	True
Aggregates	Aggregate *Subscription* – MinimumActualTime2	True
Aggregates	Aggregate *Subscription* – NumberOfTransitions	True
Aggregates	Aggregate *Subscription* – PercentBad	True
Aggregates	Aggregate *Subscription* – PercentGood	True
Aggregates	Aggregate *Subscription* – Range	True
Aggregates	Aggregate *Subscription* – Range2	True
Aggregates	Aggregate *Subscription* – StandardDeviationPopulation	True
Aggregates	Aggregate *Subscription* – StandardDeviationSample	True
Aggregates	Aggregate *Subscription* – Start	True
Aggregates	Aggregate *Subscription* – StartBound	True
Aggregates	Aggregate *Subscription* – TimeAverage	True
Aggregates	Aggregate *Subscription* – TimeAverage2	True
Aggregates	Aggregate *Subscription* – Total	True
Aggregates	Aggregate *Subscription* – Total2	True
Aggregates	Aggregate *Subscription* – VariancePopulation	True
Aggregates	Aggregate *Subscription* – VarianceSample	True
Aggregates	Aggregate *Subscription* – WorstQuality	True
Aggregates	Aggregate *Subscription* – WorstQuality2	True
Monitored Item *Services*	Monitor Aggregate Filter	False

6.6.64 Nano Embedded Device Server Profile

Nano Embedded Device Server Profile has been deprecated as it is superseded with Nano Embedded Device 2017 Server Profile in v1.04.

6.6.65 Nano Embedded Device 2017 Server Profile

Table 82 describes the details of the Nano Embedded Device 2017 *Server Profile*. This *Profile* is a FullFeatured *Profile* intended for chip level devices with limited resources. This *Profile* is functionally equivalent to the Core *Server* Facet and defines the OPC UA TCP binary protocol as the required transport profile.The support of Diagnostic *Objects* and *Variables* is optional for this *Profile* despite it being defined as "mandatory" in IEC 62541-5. Support of Diagnostic *Objects* and *Variables* is mandatory in some higher-level *Profiles*. Exposing types in the *AddressSpace* is optional for this *Profile* except if custom types (i.e. types that are derived from well-known *ObjectTypes*, VariableTypes, *ReferenceType* or DataTypes) are used. Exposing all supported types in the *AddressSpace* is mandatory in some higher-level *Profiles*. This profile supersedes the "Nano Embedded Device *Server Profile*".

Table 82 – Nano Embedded Device 2017 Server Profile

Group	Conformance Unit / *Profile* Title	Optional
Profile	Core 2017 *Server* Facet	False
Profile	UA-TCP UA-SC UA-Binary	False
Base Information	Base Info Custom Type System	True
Base Information	Base Info Diagnostics	True

6.6.66 Micro Embedded Device Server Profile

Micro Embedded Device Server Profile has been deprecated as it is superseded with Micro Embedded Device 2017 Server Profile in v1.04.

6.6.67 Micro Embedded Device 2017 Server Profile

Table 83 describes the details of the Micro Embedded Device 2017 *Server Profile*. This *Profile* is a FullFeatured *Profile* intended for small devices with limited resources. This *Profile* builds upon the Nano Embedded Device *Server Profile*. The most important additions are: support for subscriptions via the Embedded Data Change *Subscription Server* Facet and support for at least two sessions. A complete Type System is not required; however, if the *Server* implements any non-UA types, then these types and their super-types shall be exposed.
This profile supersedes the "Micro Embedded Device *Server Profile*".

Table 83 – Micro Embedded Device 2017 Server Profile

Group	Conformance Unit / *Profile* Title	Optional
Profile	Embedded DataChange *Subscription Server* Facet	False
Profile	Nano Embedded Device 2017 *Server Profile*	False
Session Services	*Session* Minimum 2 Parallel	False

6.6.68 Embedded UA Server Profile

Embedded UA Server Profile has been deprecated as it is superseded with Embedded 2017 UA Server Profile in v1.04.

6.6.69 Embedded 2017 UA Server Profile

Table 84 describes the details of the Embedded 2017 UA *Server Profile*. This *Profile* is a FullFeatured *Profile* that is intended for devices with more than 50 MB of memory and a more powerful processor. This *Profile* builds upon the Micro Embedded Device *Server Profile*. The most important additions are: support for security via the Security Policies and support for the Standard DataChange *Subscription Server* Facet. This *Profile* also requires that Servers expose all OPC-UA types that are used by the *Server* including their components and their super-types. This profile supersedes the "Embedded Device *Server Profile*".

Table 84 – Embedded 2017 UA Server Profile

Group	Conformance Unit / *Profile* Title	Optional
Profile	Micro Embedded Device 2017 *Server Profile*	False
Profile	Standard DataChange *Subscription* 2017 *Server* Facet	False
Base Information	Base Info Engineering Units	True
Base Information	Base Info Type System	False
Security	Security – No Application Authentication	True
Security	Security Default ApplicationInstance Certificate	False
Security	Security Policy Required	False

6.6.70 Standard UA Server Profile

Standard UA Server Profile has been deprecated as it is superseded with Standard 2017 UA Server Profile in v1.04.

6.6.71 Standard 2017 UA Server Profile

Table 85 describes the details of the Standard 2017 UA *Server Profile*. This *Profile* is a FullFeatured *Profile* that defines a minimum set of functionality required for PC based OPC UA servers. Compared to the embedded profiles, the *Profile* requires higher limits for Sessions, Subscriptions and Monitored Items. It also requires support of diagnostic information. This profile supersedes the "Standard UA *Server Profile*".

Table 85 – Standard 2017 UA Server Profile

Group	Conformance Unit / *Profile* Title	Optional
Profile	Embedded 2017 UA *Server Profile*	False
Profile	Enhanced DataChange *Subscription* 2017 *Server* Facet	False
Profile	User Token – X509 *Certificate Server* Facet	False
Attribute Services	*Attribute* Write StatusCode & Timestamp	True
Base Information	Base Info Diagnostics	False
Discovery Services	*Discovery* Register	False
Discovery Services	*Discovery* Register2	False
Session Services	*Session* Cancel	False
Session Services	*Session* Change User	True
Session Services	*Session* Minimum 50 Parallel	False
View *Services*	View Minimum Continuation Point 05	False

6.6.72 Core Client Facet

Core Client Facet has been deprecated as it is superseded with Core 2017 Client Facet in v1.04.

6.6.73 Core 2017 Client Facet

Table 86 describes the details of the Core 2017 *Client* Facet. This Facet defines the core functionality required for any *Client*. This Facet includes the core functions for Security and *Session* handling.

This Facet supersedes the Core *Client* Facet.

Table 86 – Core 2017 Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	SecurityPolicy – None	False
Profile	User Token – User Name Password *Client* Facet	False
Profile	User Token – X509 *Certificate Client* Facet	False
Base Information	Base Info *Client* Estimated Return Time	True
Base Information	Base Info *Client* Selection List	True
Security	Security Administration	False
Security	Security Policy Required	False
Session Services	*Session Client* Auto Reconnect	False
Session Services	*Session Client* Base	False
Session Services	*Session Client* Cancel	True
Session Services	*Session Client* Detect Shutdown	False
Session Services	*Session Client* General *Service* Behaviour	False
Session Services	*Session Client* Impersonate	True
Session Services	*Session Client* KeepAlive	False
Session Services	*Session Client* Renew NodeIds	True

6.6.74 Sessionless Client Facet

Table 87 describes the details of the Sessionless *Client* Facet. Defines the use of Sessionless *Service* invocation in a *Client*.

Table 87 – Sessionless Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Discovery Services	*Discovery Client* Get Endpoints SessionLess	True
Session Services	*Session Client* SessionLess *Service* Calls	False

6.6.75 Reverse Connect Client Facet

Table 88 describes the details of the Reverse Connect *Client* Facet. This Facet defines support of reverse connectivity in a *Client*. Usually, a connection is opened by the *Client* before starting the UA-specific handshake. This will fail, however, when *Servers* are behind firewalls. In the reverse connectivity scenario, the *Client* accepts a connection request and a ReverseHello message from a *Server* and establishes a Secure Channel using this connection.

Table 88 – Reverse Connect Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Protocol and Encoding	Protocol Reverse Connect *Client*	False

6.6.76 Base Client Behaviour Facet

Table 89 describes the details of the Base *Client* Behaviour Facet. This Facet indicates that the *Client* supports behaviour that *Clients* shall follow for best use by operators and administrators. They include allowing configuration of an endpoint for a server without using the discovery service set; Support for manual security setting configuration and behaviour with regard to security issues; support for Automatic reconnection to a disconnected server. These behaviours can only be tested in a test lab. They are best practice guidelines.

Table 89 – Base Client Behaviour Facet

Group	Conformance Unit / *Profile* Title	Optional
Base Information	Base Info *Client* Remote *Nodes*	True
Discovery Services	*Discovery Client* Configure Endpoint	False
Security	Security Administration	False
Security	Security Administration – XML Schema	False
Security	Security *Certificate* Administration	False
Session Services	*Session Client* Auto Reconnect	True
Subscription Services	*Subscription Client* Multiple	False
Subscription Services	*Subscription Client* Publish Configurable	False

6.6.77 Discovery Client Facet

Table 90 describes the details of the *Discovery Client* Facet. This Facet defines the ability to discover *Servers* and their Endpoints.

Table 90 – Discovery Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Discovery Services	*Discovery Client* Configure Endpoint	False
Discovery Services	*Discovery Client* Find *Servers* Basic	False
Discovery Services	*Discovery Client* Find *Servers* Dynamic	False
Discovery Services	*Discovery Client* Find *Servers* with URI	True
Discovery Services	*Discovery Client* Get Endpoints Basic	False
Discovery Services	*Discovery Client* Get Endpoints Dynamic	False

6.6.78 Subnet Discovery Client Facet

Table 91 describes the details of the Subnet *Discovery Client* Facet. Support of this Facet enables discovery of the *Server* on a subnet.

Table 91 – Subnet Discovery Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Discovery Services	*Discovery Client* Find *Servers* on Network	False
Discovery Services	*Discovery Client* Find *Servers* on Network using LDS-ME	True
Discovery Services	*Discovery Client* Find *Servers* on Network using mDNS	True

6.6.79 Global Discovery Client Facet

Table 92 describes the details of the Global *Discovery Client* Facet. Support of this Facet enables system-wide discovery of *Servers* using a Global *Discovery Server* (GDS).

Table 92 – Global Discovery Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Discovery Services	*Discovery Client* Find Applications in GDS	True
Discovery Services	*Discovery Client* Find *Servers* in GDS	False

6.6.80 Global Certificate Management Client Facet

Table 93 describes the details of the Global *Certificate* Management *Client* Facet. This Facet defines the capability to interact with a Global *Certificate* Management *Server* to obtain an initial or renewed *Certificate* and Trust Lists.

Table 93 – Global Certificate Management Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Security	Pull Model for Global Certificate and TrustList Management	False

6.6.81 KeyCredential Service Client Facet

Table 94 describes the details of the KeyCredential *Service Client* Facet. This Facet defines the capability to interact with a KeyCredential *Service* to obtain KeyCredentials. For example KeyCredentials are needed to access an Authorization *Service* or a Broker. The KeyCredential *Service* is typically part of a system-wide tool, like a GDS that also manages Applications, Access Tokens, and *Certificates*.

Table 94 – KeyCredential Service Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Security	Pull Model for KeyCredential *Service*	False

6.6.82 Access Token Request Client Facet

Table 95 describes the details of the Access Token Request *Client* Facet. A *Client* Facet for using the RequestAccessToken *Method* on an Authorization *Server* (defined in IEC 62541-12) to request such a token.

Table 95 – Access Token Request Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Security	Authorization *Service Client*	False

6.6.83 AddressSpace Lookup Client Facet

Table 96 describes the details of the *AddressSpace* Lookup *Client* Facet. This Facet defines the ability to navigate through the *AddressSpace* and includes basic *AddressSpace* concepts, view and browse functionality and simple attribute read functionality.

Table 96 – AddressSpace Lookup Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Address Space Model	Address Space *Client* Base	False
Attribute Services	*Attribute Client* Read Base	False
Attribute Services	*Attribute Client* Remote *Nodes Attribute* Access	True
Base Information	Base Info *Client* Basic	False
Base Information	Base Info *Client* GetMonitoredItems *Method*	True
View *Services*	View *Client* Basic Browse	False
View *Services*	View *Client* Basic ResultSet Filtering	False
View *Services*	View *Client* RegisterNodes	True
View *Services*	View *Client* Remote *Nodes* Browse	True
View *Services*	View *Client* Remote *Nodes* Translate Browse	True
View *Services*	View *Client* TranslateBrowsePath	True

6.6.84 Request State Change Client Facet

Table 97 describes the details of the Request State Change *Client* Facet. This Facet specifies the ability to invoke the RequestServerStateChange *Method*.

Table 97 – Request State Change Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Base Information	Base Info *Client* RequestServerStateChange	False

6.6.85 File Access Client Facet

Table 98 describes the details of the File Access *Client* Facet. This Facet defines the ability to use File transfer via the defined FileType. This includes reading and optionally writing.

Table 98 – File Access Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Base Information	Base Info *Client* FileType Base	False
Base Information	Base Info *Client* FileType Write	True

6.6.86 Entry Level Support 2015 Client Facet

Table 99 describes the details of the Entry Level Support 2015 *Client* Facet. This Facet defines the ability to interoperate with low-end *Servers*, in particular *Servers* that support the Nano Embedded *Profile* but in general *Servers* with defined limits.

Table 99 – Entry Level Support 2015 Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Base Information	Base Info *Client* Honour Operation Limits	False
Base Information	Base Info *Client* Type Pre-Knowledge	False
Session Services	*Session Client* Single *Session*	False
Subscription Services	*Subscription Client* Fallback	False

6.6.87 Multi-Server Client Connection Facet

Table 100 describes the details of the Multi-*Server Client* Connection Facet. This Facet defines the ability for simultaneous access to multiple *Servers*.

Table 100 – Multi-Server Client Connection Facet

Group	Conformance Unit / *Profile* Title	Optional
Session Services	*Session Client* Multiple Connections	False

6.6.88 Documentation – Client

Table 101 describes the details of the Documentation – *Client*. This Facet provides a list of user documentation that a *Client* application should provide.

Table 101 – Documentation – Client

Group	Conformance Unit / *Profile* Title	Optional
Miscellaneous	Documentation *Client* – Installation	False
Miscellaneous	Documentation *Client* – Multiple Languages	True
Miscellaneous	Documentation *Client* – On-line	True
Miscellaneous	Documentation *Client* – Supported *Profiles*	True
Miscellaneous	Documentation *Client* – Trouble Shooting Guide	True
Miscellaneous	Documentation *Client* – Users Guide	False

6.6.89 Attribute Read Client Facet

Table 102 describes the details of the *Attribute* Read *Client* Facet. This Facet defines the ability to read *Attribute* values of *Nodes*.

Table 102 – Attribute Read Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Address Space Model	Address Space *Client* Atomicity	True
Address Space Model	Address Space *Client* Complex Data Dictionary	True
Address Space Model	Address Space *Client* DataTypeDefinition *Attribute*	True
Address Space Model	Address Space *Client* Full Array Only	True
Attribute Services	*Attribute Client* Read Base	False
Attribute Services	*Attribute Client* Read Complex	True
Attribute Services	*Attribute Client* Read with proper Encoding	True

6.6.90 Attribute Write Client Facet

Table 103 describes the details of the *Attribute* Write *Client* Facet. This Facet defines the ability to write *Attribute* values of *Nodes*.

Table 103 – Attribute Write Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Address Space Model	Address Space *Client* Atomicity	True
Address Space Model	Address Space *Client* Complex Data Dictionary	True
Address Space Model	Address Space *Client* DataTypeDefinition *Attribute*	True
Address Space Model	Address Space *Client* Full Array Only	True
Attribute Services	*Attribute Client* Write Base	False
Attribute Services	*Attribute Client* Write Complex	True
Attribute Services	*Attribute Client* Write Quality & Timestamp	True

6.6.91 DataChange Subscriber Client Facet

Table 104 describes the details of the DataChange Subscriber *Client* Facet. This Facet defines the ability to monitor *Attribute* values for data change.

Table 104 – DataChange Subscriber Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Address Space Model	Address Space *Client* Atomicity	True
Address Space Model	Address Space *Client* Complex Data Dictionary	True
Address Space Model	Address Space *Client* DataTypeDefinition *Attribute*	True
Address Space Model	Address Space *Client* Full Array Only	True
Base Information	Base Data *Client* ResendData *Method*	True
Base Information	Base Info *Client* GetMonitoredItems *Method*	True
Monitored Item *Services*	Monitor *Client* by Index	False
Monitored Item *Services*	Monitor *Client* Complex Value	True
Monitored Item *Services*	Monitor *Client* Deadband Filter	True
Monitored Item *Services*	Monitor *Client* Modify	True
Monitored Item *Services*	Monitor *Client* Trigger	True
Monitored Item *Services*	Monitor *Client* Value Change	False
Subscription Services	*Subscription Client* Basic	False
Subscription Services	*Subscription Client* Modify	True
Subscription Services	*Subscription Client* Multiple	True
Subscription Services	*Subscription Client* Republish	False

6.6.92 Durable Subscription Client Facet

Table 105 describes the details of the Durable *Subscription Client* Facet. This Facet specifies use of durable Subscriptions. It implies support of any of the DataChange or *Event* Subscriber Facets.

Table 105 – Durable Subscription Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Subscription Services	*Subscription Client* Durable	False

6.6.93 DataAccess Client Facet

Table 106 describes the details of the DataAccess *Client* Facet. This Facet defines the ability to utilize the DataAccess Information Model, i.e., industrial automation data like analogue and discrete data items and their quality of service.

Table 106 – DataAccess Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Address Space Model	Address Space *Client* Base	False
Address Space Model	Address Space *Client* Complex Data Dictionary	True
Attribute Services	*Attribute Client* Read Base	False
Attribute Services	*Attribute Client* Read Complex	True
Attribute Services	*Attribute Client* Read with proper Encoding	True
Data Access	Data Access *Client* AnalogItems	True
Data Access	Data Access *Client* Basic	False
Data Access	Data Access *Client* Deadband	True
Data Access	Data Access *Client* MultiState	True
Data Access	Data Access *Client* MultiStateValueDiscrete	True
Data Access	Data Access *Client* SemanticChange	True
Data Access	Data Access *Client* TwoState	True

6.6.94 Event Subscriber Client Facet

Table 107 describes the details of the *Event* Subscriber *Client* Facet. This Facet defines the ability to subscribe for *Event Notifications*. This includes basic *AddressSpace* concept and the browsing of it, adding events and event filters as monitored items and adding subscriptions.

Table 107 – Event Subscriber Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Address Space Model	Address Space *Client* Base	False
Monitored Item *Services*	Monitor *Client* Complex *Event* Filter	True
Monitored Item *Services*	Monitor *Client Event* Filter	False
Monitored Item *Services*	Monitor *Client* Events	False
Monitored Item *Services*	Monitor *Client* Modify	True
Monitored Item *Services*	Monitor *Client* Trigger	True
Subscription Services	*Subscription Client* Basic	False
Subscription Services	*Subscription Client* Modify	True
Subscription Services	*Subscription Client* Multiple	True
Subscription Services	*Subscription Client* Republish	False
View *Services*	View *Client* Basic Browse	True
View *Services*	View *Client* TranslateBrowsePath	True

6.6.95 Base Event Processing Client Facet

Table 108 describes the details of the Base *Event* Processing *Client* Facet. This Facet defines the ability to subscribe for and process basic OPC UA Events. The *Client* has to support at least one of the Events in the Facet.

Table 108 – Base Event Processing Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	*Event* Subscriber *Client* Facet	False
Base Information	Base Info *Client* Change Events	True
Base Information	Base Info *Client* Device Failure	True
Base Information	Base Info *Client* Progress Events	True
Base Information	Base Info *Client* System Status	True
Base Information	Base Info *Client* System Status Underlying System	True
Base Information	Base Info *Event* Processing	False

6.6.96 Notifier and Source Hierarchy Client Facet

Table 109 describes the details of the Notifier and Source Hierarchy *Client* Facet. This Facet defines the ability to find and use a hierarchy of *Objects* that are event notifier and *Nodes* that are event sources in the *Server AddressSpace*.

Table 109 – Notifier and Source Hierarchy Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	*Event* Subscriber *Client* Facet	False
Address Space Model	Address Space *Client* Notifier Hierarchy	False
Address Space Model	Address Space *Client* Source Hierarchy	False
Subscription Services	*Subscription Client* Publish Configurable	False

6.6.97 A & C Base Condition Client Facet

Table 110 describes the details of the A & C Base *Condition Client* Facet. This Facet defines the ability to use the *Alarm* and *Condition* basic model. This includes the ability to subscribe for Events and to initiate a Refresh *Method*.

Table 110 – A & C Base Condition Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	*Event* Subscriber *Client* Facet	False
Profile	*Method Client* Facet	False
Alarms and *Conditions*	A & C Basic *Client*	False
Alarms and *Conditions*	A & C *Condition* Sub-Classes *Client*	True
Alarms and *Conditions*	A & C ConditionClasses *Client*	False
Alarms and *Conditions*	A & C Refresh *Client*	False

6.6.98 A & C Refresh2 Client Facet

Table 111 describes the details of the A & C Refresh2 *Client* Facet. This Facet enhances the A & C Base *Condition Server* Facet with the ability to initiate a ConditionRefresh2 *Method*.

Table 111 – A & C Refresh2 Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	A & C Base *Condition Client* Facet	False
Alarms and *Conditions*	A & C Refresh2 *Client*	False

6.6.99 A & C Address Space Instance Client Facet

Table 112 describes the details of the A & C Address Space Instance *Client* Facet. This Facet defines the ability to use *Condition* instances in the *AddressSpace*.

Table 112 – A & C Address Space Instance Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Alarms and *Conditions*	A & C Instances *Client*	False

6.6.100 A & C Enable Client Facet

Table 113 describes the details of the A & C Enable *Client* Facet. This Facet defines the ability to enable and disable *Alarms*.

Table 113 – A & C Enable Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	A & C Base *Condition Client* Facet	False
Alarms and *Conditions*	A & C Enable *Client*	False

6.6.101 A & C AlarmMetrics Client Facet

Table 114 describes the details of the A & C AlarmMetrics *Client* Facet. This Facet defines the ability to use the AlarmMetrics model, i.e. understand and use the collected alarm metrics at any level in the HasNotifier hierarchy.

Table 114 – A & C AlarmMetrics Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Alarms and *Conditions*	A & C *Alarm* Metrics *Client*	False

6.6.102 A & C Alarm Client Facet

Table 115 describes the details of the A & C *Alarm Client* Facet. This Facet defines the ability to use the alarming model (the AlarmType or any of the sub-types).

Table 115 – A & C Alarm Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	A & C Base *Condition Client* Facet	False
Alarms and *Conditions*	A & C Acknowledge *Client*	False
Alarms and *Conditions*	A & C *Alarm Client*	False
Alarms and *Conditions*	A & C Audible Sound *Client*	True
Alarms and *Conditions*	A & C Comment *Client*	True
Alarms and *Conditions*	A & C Confirm *Client*	True
Alarms and *Conditions*	A & C Discrepancy Client	True
Alarms and *Conditions*	A & C Discrete *Client*	False
Alarms and *Conditions*	A & C First in Group *Alarm Client*	True
Alarms and *Conditions*	A & C OffNormal *Client*	True
Alarms and *Conditions*	A & C On-Off Delay *Client*	True
Alarms and *Conditions*	A & C Out Of *Service Client*	True
Alarms and *Conditions*	A & C Re-Alarming *Client*	True
Alarms and *Conditions*	A & C Shelving *Client*	True
Alarms and Conditions	A & C Silencing Client	True
Alarms and *Conditions*	A & C Suppression by Operator *Client*	True
Alarms and Conditions	A & C Suppression Client	True
Alarms and *Conditions*	A & C SystemOffNormal *Client*	True
Alarms and *Conditions*	A & C Trip *Client*	True

6.6.103 A & C Exclusive Alarming Client Facet

Table 116 describes the details of the A & C Exclusive Alarming *Client* Facet. This Facet defines the ability to use the exclusive *Alarm* model. This includes understanding the various subtypes such as ExclusiveRateOfChangeAlarm, ExclusiveLevelAlarm and ExclusiveDeviationAlarm.

Table 116 – A & C Exclusive Alarming Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	A & C *Alarm Client* Facet	False
Alarms and *Conditions*	A & C Exclusive Deviation *Client*	True
Alarms and *Conditions*	A & C Exclusive Level *Client*	True
Alarms and *Conditions*	A & C Exclusive Limit *Client*	False
Alarms and *Conditions*	A & C Exclusive RateOfChange *Client*	True

6.6.104 A & C Non-Exclusive Alarming Client Facet

Table 117 describes the details of the A & C Non-Exclusive Alarming *Client* Facet. This Facet defines the ability to use the non-exclusive *Alarm* model. This includes understanding the various subtypes such as NonExclusiveRateOfChangeAlarm, NonExclusiveLevelAlarm and NonExclusiveDeviationAlarm.

Table 117 – A & C Non-Exclusive Alarming Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	A & C *Alarm Client* Facet	False
Alarms and *Conditions*	A & C Non-Exclusive Deviation *Client*	True
Alarms and *Conditions*	A & C Non-Exclusive Level *Client*	True
Alarms and *Conditions*	A & C Non-Exclusive Limit *Client*	False
Alarms and *Conditions*	A & C Non-Exclusive RateOfChange *Client*	True

6.6.105 A & C Previous Instances Client Facet

Table 118 describes the details of the A & C Previous Instances *Client* Facet. This Facet defines the ability to use previous instances of *Alarms*. This implies the ability to understand branchIds.

Table 118 – A & C Previous Instances Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	A & C Base *Condition Client* Facet	False
Alarms and *Conditions*	A & C Branch *Client*	False

6.6.106 A & C Dialog Client Facet

Table 119 describes the details of the A & C Dialog *Client* Facet. This Facet defines the ability to use the dialog model. This implies the support of *Method* invocation to respond to dialog messages.

Table 119 – A & C Dialog Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	A & C Base *Condition Client* Facet	False
Alarms and *Conditions*	A & C Dialog *Client*	False

6.6.107 A & C CertificateExpiration Client Facet

Table 120 describes the details of the A & C CertificateExpiration *Client* Facet. This Facet defines the ability to use the CertificateExpirationAlarmType.

Table 120 – A & C CertificateExpiration Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	A & C Base *Condition Client* Facet	False
Alarms and *Conditions*	A & C Acknowledge *Client*	True
Alarms and *Conditions*	A & C *Alarm Client*	False
Alarms and *Conditions*	A & C CertificateExpiration *Client*	False
Alarms and *Conditions*	A & C Comment *Client*	True
Alarms and *Conditions*	A & C Confirm *Client*	True
Alarms and *Conditions*	A & C Shelving *Client*	True

6.6.108 A & E Proxy Facet

Table 121 describes the details of the A & E Proxy Facet. This Facet describes the functionality used by a default A & E *Client* proxy. A *Client* exposes this Facet so that a *Server* may be able to better understand the commands that are being issued by the *Client*, since this Facet indicates that the *Client* is an A&E Com *Client*.

Table 121 – A & E Proxy Facet

Group	Conformance Unit / *Profile* Title	Optional
Address Space Model	Address Space *Client* Base	False
Alarms and *Conditions*	A & C Acknowledge *Client*	False
Alarms and *Conditions*	A & C *Alarm Client*	False
Alarms and *Conditions*	A & C Basic *Client*	False
Alarms and *Conditions*	A & C ConditionClasses *Client*	False
Alarms and *Conditions*	A & C Discrete *Client*	False
Alarms and *Conditions*	A & C Exclusive Deviation *Client*	False
Alarms and *Conditions*	A & C Exclusive Level *Client*	False
Alarms and *Conditions*	A & C Exclusive Limit *Client*	False
Alarms and *Conditions*	A & C Exclusive RateOfChange *Client*	False
Alarms and *Conditions*	A & C Instances *Client*	False
Alarms and *Conditions*	A & C Non-Exclusive Deviation *Client*	False
Alarms and *Conditions*	A & C Non-Exclusive Level *Client*	False
Alarms and *Conditions*	A & C Non-Exclusive Limit *Client*	False
Alarms and *Conditions*	A & C Non-Exclusive RateOfChange *Client*	False
Alarms and *Conditions*	A & C OffNormal *Client*	False
Alarms and *Conditions*	A & C Refresh *Client*	False
Alarms and *Conditions*	A & C SystemOffNormal *Client*	True
Alarms and *Conditions*	A & C Trip *Client*	False
Attribute Services	*Attribute Client* Read Base	False
Base Information	Base Info *Client* Basic	False
Base Information	Base Info *Client* Change Events	False
Discovery Services	*Discovery Client* Configure Endpoint	False
Discovery Services	*Discovery Client* Find *Servers* Basic	False
Discovery Services	*Discovery Client* Find *Servers* Dynamic	False
Discovery Services	*Discovery Client* Find *Servers* with URI	False
Discovery Services	*Discovery Client* Get Endpoints Basic	False
Discovery Services	*Discovery Client* Get Endpoints Dynamic	False
Method Services	*Method Client* Call	False
Monitored Item *Services*	Monitor *Client* Complex *Event* Filter	False
Monitored Item *Services*	Monitor *Client Event* Filter	False
Monitored Item *Services*	Monitor *Client* Events	False
Security	Security Administration	False
Security	Security Administration – XML Schema	False
Security	Security *Certificate* Administration	False
Session Services	*Session Client* Auto Reconnect	False
Subscription Services	*Subscription Client* Basic	False

Group	Conformance Unit / *Profile* Title	Optional
Subscription Services	*Subscription Client* Multiple	False
Subscription Services	*Subscription Client* Publish Configurable	False
Subscription Services	*Subscription Client* Republish	False
View *Services*	View *Client* Basic Browse	False
View *Services*	View *Client* Basic ResultSet Filtering	False
View *Services*	View *Client* TranslateBrowsePath	False

6.6.109 Method Client Facet

Table 122 describes the details of the *Method Client* Facet. This Facet defines the ability to call arbitrary Methods.

Table 122 – Method Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Method Services	*Method Client* Call	False

6.6.110 Auditing Client Facet

Table 123 describes the details of the Auditing *Client* Facet. This Facet defines the ability to monitor Audit Events.

Table 123 – Auditing Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	*Event* Subscriber *Client* Facet	False
Auditing	Auditing *Client* Audit ID	False
Auditing	Auditing *Client* Subscribes	False

6.6.111 Node Management Client Facet

Table 124 describes the details of the *Node* Management *Client* Facet. This Facet defines the ability to configure the *AddressSpace* of an OPC UA *Server* through OPC UA *Node* Management *Service* Set.

Table 124 – Node Management Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Address Space Model	Address Space *Client* Base	False
Node Management *Services*	*Node* Management *Client*	False

6.6.112 Advanced Type Programming Client Facet

Table 125 describes the details of the Advanced Type Programming *Client* Facet. This Facet defines the ability to use the type model and process the instance *AddressSpace* based on the type model. For example a client may contain generic displays that are based on a type, in that they contain a relative path from some main type. On call up this main type is matched to an instance and all of display items are resolved based on the provided type model.

Table 125 – Advanced Type Programming Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Address Space Model	Address Space *Client* Base	False
Base Information	Base Info *Client* Basic	False
Base Information	Base Info *Client* Type Programming	False
View *Services*	View *Client* TranslateBrowsePath	False

6.6.113 User Role Management Client Facet

Table 126 describes the details of the User Role Management *Client Facet*. This *Facet* defines knowledge of the OPC UA *Information Model* for user roles and permissions and the use of the *Methods* to manage them.

Table 126 – User Role Management Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Security	Security Role *Client* Base	False
Security	Security Role *Client* DefaultRolePermissions	False
Security	Security Role *Client* Management	False
Security	Security Role *Client* Restrict Applications	True
Security	Security Role *Client* Restrict Endpoints	True
Security	Security Role *Client* RolePermissions	False

6.6.114 State Machine Client Facet

Table 127 describes the details of the State Machine *Client* Facet. This Facet defines the ability to use state machines based on the StateMachineType or a sub-type.

Table 127 – State Machine Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Base Information	Base Info Client Available States and Transitions	True
Base Information	Base Info *Client* Finite State Machine Instance	True
Base Information	Base Info *Client* State Machine Instance	False

6.6.115 Diagnostic Client Facet

Table 128 describes the details of the Diagnostic *Client* Facet. This Facet defines the ability to read and process diagnostic information that is part of the OPC UA information model.

Table 128 – Diagnostic Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Address Space Model	Address Space *Client* Base	False
Base Information	Base Info *Client* Basic	False
Base Information	Base Info *Client* Diagnostics	False

6.6.116 Redundant Client Facet

Table 129 describes the details of the Redundant *Client* Facet. This Facet defines the ability to use the redundancy feature available for redundant *Clients*.

Table 129 – Redundant Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Redundancy	Redundancy *Client*	False
Subscription Services	*Subscription Client* TransferSubscriptions	True

6.6.117 Redundancy Switch Client Facet

Table 130 describes the details of the Redundancy Switch *Client* Facet. A *Client* that supports this Facet supports monitoring the redundancy status for non-transparent redundant *Servers* and switching to the backup *Server* when they recognize a change.

Table 130 – Redundancy Switch Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Redundancy	Redundancy *Client* Switch	False

6.6.118 Historical Access Client Facet

Table 131 describes the details of the Historical Access *Client* Facet. This Facet defines the ability to read, process, and update historical data.

Table 131 – Historical Access Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Attribute Services	*Attribute Client* Historical Read	False
Historical Access	Historical Access *Client* Browse	False
Historical Access	Historical Access *Client* Read Raw	False

6.6.119 Historical Data AtTime Client Facet

Table 132 describes the details of the Historical Data AtTime *Client* Facet. This Facet defines the ability to access data at specific instances in time.

Table 132 – Historical Data AtTime Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	Historical Access *Client* Facet	False
Historical Access	Historical Access *Client* Time Instance	False

6.6.120 Historical Aggregate Client Facet

Table 133 describes the details of the Historical Aggregate *Client* Facet. This Facet defines the ability to read historical data by specifying the needed aggregate. This implies consideration of the list of aggregates supported by the *Server*.

Table 133 – Historical Aggregate Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Aggregates	Aggregate – *Client* AnnotationCount	True
Aggregates	Aggregate – *Client* Average	True
Aggregates	Aggregate – *Client* Count	True
Aggregates	Aggregate – *Client* Custom Aggregates	True
Aggregates	Aggregate – *Client* Delta	True
Aggregates	Aggregate – *Client* DeltaBounds	True
Aggregates	Aggregate – *Client* DurationBad	True
Aggregates	Aggregate – *Client* DurationGood	True
Aggregates	Aggregate – *Client* DurationInStateNonZero	True
Aggregates	Aggregate – *Client* DurationInStateZero	True
Aggregates	Aggregate – *Client* End	True
Aggregates	Aggregate – *Client* EndBound	True
Aggregates	Aggregate – *Client* Interpolative	True
Aggregates	Aggregate – *Client* Maximum	True
Aggregates	Aggregate – *Client* Maximum2	True
Aggregates	Aggregate – *Client* MaximumActualTime	True
Aggregates	Aggregate – *Client* MaximumActualTime2	True
Aggregates	Aggregate – *Client* Minimum	True
Aggregates	Aggregate – *Client* Minimum2	True
Aggregates	Aggregate – *Client* MinimumActualTime	True
Aggregates	Aggregate – *Client* MinimumActualTime2	True
Aggregates	Aggregate – *Client* NumberOfTransitions	True
Aggregates	Aggregate – *Client* PercentBad	True
Aggregates	Aggregate – *Client* PercentGood	True
Aggregates	Aggregate – *Client* Range	True
Aggregates	Aggregate – *Client* Range2	True
Aggregates	Aggregate – *Client* StandardDeviationPopulation	True
Aggregates	Aggregate – *Client* StandardDeviationSample	True
Aggregates	Aggregate – *Client* Start	True
Aggregates	Aggregate – *Client* StartBound	True
Aggregates	Aggregate – *Client* TimeAverage	True
Aggregates	Aggregate – *Client* TimeAverage2	True
Aggregates	Aggregate – *Client* Total	True
Aggregates	Aggregate – *Client* Total2	True
Aggregates	Aggregate – *Client* Usage	False
Aggregates	Aggregate – *Client* VariancePopulation	True
Aggregates	Aggregate – *Client* VarianceSample	True
Aggregates	Aggregate – *Client* WorstQuality	True
Aggregates	Aggregate – *Client* WorstQuality2	True
Historical Access	Historical Access *Client* Read Aggregates	False

6.6.121 Historical Annotation Client Facet

Table 134 describes the details of the Historical Annotation *Client* Facet. This Facet defines the ability to retrieve and write annotations for historical data.

Table 134 – Historical Annotation Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	Historical Access *Client* Facet	False
Profile	Historical Data Update *Client* Facet	False
Historical Access	Historical Access *Client* Annotations	False

6.6.122 Historical Access Modified Data Client Facet

Table 135 describes the details of the Historical Access Modified Data *Client* Facet. This Facet defines the ability to access prior historical data (values that were modified or inserted).

Table 135 – Historical Access Modified Data Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	Historical Access *Client* Facet	False
Historical Access	Historical Access *Client* Read Modified	False

6.6.123 Historical Data Insert Client Facet

Table 136 describes the details of the Historical Data Insert *Client* Facet. This Facet defines the ability to insert historical data.

Table 136 – Historical Data Insert Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Attribute Services	*Attribute Client* Historical Updates	False
Historical Access	Historical Access *Client* Data Insert	False

6.6.124 Historical Data Update Client Facet

Table 137 describes the details of the Historical Data Update *Client* Facet. This Facet defines the ability to update historical data.

Table 137 – Historical Data Update Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Attribute Services	*Attribute Client* Historical Updates	False
Historical Access	Historical Access *Client* Data Update	False

6.6.125 Historical Data Replace Client Facet

Table 138 describes the details of the Historical Data Replace *Client* Facet. This Facet defines the ability to replace historical data.

Table 138 – Historical Data Replace Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Attribute Services	*Attribute Client* Historical Updates	False
Historical Access	Historical Access *Client* Data Replace	False

6.6.126 Historical Data Delete Client Facet

Table 139 describes the details of the Historical Data Delete *Client* Facet. This Facet defines the ability to delete historical data.

Table 139 – Historical Data Delete Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Attribute Services	*Attribute Client* Historical Updates	False
Historical Access	Historical Access *Client* Data Delete	False

6.6.127 Historical Access Client Server Timestamp Facet

Table 140 describes the details of the Historical Access *Client Server* Timestamp Facet. This Facet defines the ability to request and process *Server* timestamps, in addition to source timestamps.

Table 140 – Historical Access Client Server Timestamp Facet

Group	Conformance Unit / *Profile* Title	Optional
Historical Access	Historical Access *Client Server* Timestamp	False

6.6.128 Historical Structured Data Access Client Facet

Table 141 describes the details of the Historical Structured Data Access *Client* Facet. This Facet defines the ability to read structured values for historical nodes.

Table 141 – Historical Structured Data Access Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	Historical Access *Client* Facet	False
Historical Access	Historical Access *Client* Structure Data Raw	False

6.6.129 Historical Structured Data AtTime Client Facet

Table 142 describes the details of the Historical Structured Data AtTime *Client* Facet. This Facet defines the ability to read structured values for historical nodes at specific instances in time.

Table 142 – Historical Structured Data AtTime Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	Historical Data AtTime *Client* Facet	False
Historical Access	Historical Access *Client* Structure Data Time Instance	False

6.6.130 Historical Structured Data Modified Client Facet

Table 143 describes the details of the Historical Structured Data Modified *Client* Facet. This Facet defines the ability to read structured values for prior historical data (values that were modified or inserted).

Table 143 – Historical Structured Data Modified Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	Historical Access Modified Data *Client* Facet	False
Historical Access	Historical Access *Client* Structure Data Read Modified	False

6.6.131 Historical Structured Data Insert Client Facet

Table 144 describes the details of the Historical Structured Data Insert *Client* Facet. This Facet defines the ability to insert structured historical data.

Table 144 – Historical Structured Data Insert Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	Historical Data Insert *Client* Facet	False
Historical Access	Historical Access *Client* Structure Data Insert	False

6.6.132 Historical Structured Data Update Client Facet

Table 145 describes the details of the Historical Structured Data Update *Client* Facet. This Facet defines the ability to update structured historical data.

Table 145 – Historical Structured Data Update Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	Historical Data Update *Client* Facet	False
Historical Access	Historical Access *Client* Structure Data Update	False

6.6.133 Historical Structured Data Replace Client Facet

Table 146 describes the details of the Historical Structured Data Replace *Client* Facet. This Facet defines the ability to replace structured historical data.

Table 146 – Historical Structured Data Replace Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	Historical Data Replace *Client* Facet	False
Historical Access	Historical Access *Client* Structure Data Replace	False

6.6.134 Historical Structured Data Delete Client Facet

Table 147 describes the details of the Historical Structured Data Delete *Client* Facet. This Facet defines the ability to remove structured historical data.

Table 147 – Historical Structured Data Delete Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	Historical Data Delete *Client* Facet	False
Historical Access	Historical Access *Client* Structure Data Delete	False

6.6.135 Historical Events Client Facet

Table 148 describes the details of the Historical Events *Client* Facet. This Facet defines the ability to read Historical Events, including simple filtering.

Table 148 – Historical Events Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Attribute Services	*Attribute Client* Historical Read	False
Historical Access	Historical Access *Client* Read Events	False

6.6.136 Historical Event Insert Client Facet

Table 149 describes the details of the Historical *Event* Insert *Client* Facet. This Facet defines the ability to insert historical events.

Table 149 – Historical Event Insert Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Attribute Services	*Attribute Client* Historical Updates	False
Historical Access	Historical Access *Client Event* Inserts	False

6.6.137 Historical Event Update Client Facet

Table 150 describes the details of the Historical *Event* Update *Client* Facet. This Facet defines the ability to update historical events.

Table 150 – Historical Event Update Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Attribute Services	*Attribute Client* Historical Updates	False
Historical Access	Historical Access *Client Event* Updates	False

6.6.138 Historical Event Replace Client Facet

Table 151 describes the details of the Historical *Event* Replace *Client* Facet. This Facet defines the ability to replace historical events.

Table 151 – Historical Event Replace Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Attribute Services	*Attribute Client* Historical Updates	False
Historical Access	Historical Access *Client Event* Replaces	False

6.6.139 Historical Event Delete Client Facet

Table 152 describes the details of the Historical *Event* Delete *Client* Facet. This Facet defines the ability to delete Historical events.

Table 152 – Historical Event Delete Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Attribute Services	*Attribute Client* Historical Updates	False
Historical Access	Historical Access *Client Event* Deletes	False

6.6.140 Aggregate Subscriber Client Facet

Table 153 describes the details of the Aggregate Subscriber *Client* Facet. This Facet defines the ability to use the aggregate filter when subscribing for *Attribute* values.

Table 153 – Aggregate Subscriber Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Aggregates	Aggregate *Subscription* – *Client* AnnotationCount	True
Aggregates	Aggregate *Subscription* – *Client* Average	True
Aggregates	Aggregate *Subscription* – *Client* Count	True
Aggregates	Aggregate *Subscription* – *Client* Custom Aggregates	True
Aggregates	Aggregate *Subscription* – *Client* Delta	True
Aggregates	Aggregate *Subscription* – *Client* DeltaBounds	True
Aggregates	Aggregate *Subscription* – *Client* DurationBad	True
Aggregates	Aggregate *Subscription* – *Client* DurationGood	True
Aggregates	Aggregate *Subscription* – *Client* DurationInStateNonZero	True
Aggregates	Aggregate *Subscription* – *Client* DurationInStateZero	True
Aggregates	Aggregate *Subscription* – *Client* End	True
Aggregates	Aggregate *Subscription* – *Client* EndBound	True
Aggregates	Aggregate *Subscription* – *Client* Filter	False
Aggregates	Aggregate *Subscription* – *Client* Interpolative	True
Aggregates	Aggregate *Subscription* – *Client* Maximum	True
Aggregates	Aggregate *Subscription* – *Client* Maximum2	True
Aggregates	Aggregate *Subscription* – *Client* MaximumActualTime	True
Aggregates	Aggregate *Subscription* – *Client* MaximumActualTime2	True
Aggregates	Aggregate *Subscription* – *Client* Minimum	True
Aggregates	Aggregate *Subscription* – *Client* Minimum2	True
Aggregates	Aggregate *Subscription* – *Client* MinimumActualTime	True
Aggregates	Aggregate *Subscription* – *Client* MinimumActualTime2	True
Aggregates	Aggregate *Subscription* – *Client* NumberOfTransitions	True
Aggregates	Aggregate *Subscription* – *Client* PercentBad	True
Aggregates	Aggregate *Subscription* – *Client* PercentGood	True
Aggregates	Aggregate *Subscription* – *Client* Range	True
Aggregates	Aggregate *Subscription* – *Client* Range2	True
Aggregates	Aggregate *Subscription* – *Client* StandardDeviationPopulation	True

Group	Conformance Unit / *Profile* Title	Optional
Aggregates	Aggregate *Subscription* – *Client* StandardDeviationSample	True
Aggregates	Aggregate *Subscription* – *Client* Start	True
Aggregates	Aggregate *Subscription* – *Client* StartBound	True
Aggregates	Aggregate *Subscription* – *Client* TimeAverage	True
Aggregates	Aggregate *Subscription* – *Client* TimeAverage2	True
Aggregates	Aggregate *Subscription* – *Client* Total	True
Aggregates	Aggregate *Subscription* – *Client* Total2	True
Aggregates	Aggregate *Subscription* – *Client* VariancePopulation	True
Aggregates	Aggregate *Subscription* – *Client* VarianceSample	True
Aggregates	Aggregate *Subscription* – *Client* WorstQuality	True
Aggregates	Aggregate *Subscription* – *Client* WorstQuality2	True
Monitored Item *Services*	Monitor *Client* Aggregate Filter	False
Monitored Item *Services*	Monitor *Client* by Index	False
Monitored Item *Services*	Monitor *Client* Modify	True
Monitored Item *Services*	Monitor *Client* Value Change	False
Subscription Services	*Subscription Client* Basic	False
Subscription Services	*Subscription Client* Modify	True
Subscription Services	*Subscription Client* Multiple	True
Subscription Services	*Subscription Client* Republish	True

6.6.141 Standard UA Client Profile

Standard UA Client Profile has been deprecated as it is superseded with Standard UA Client 2017 Profile in v1.04.

6.6.142 Standard UA Client 2017 Profile

Table 154 describes the details of the Standard UA *Client* 2017 *Profile*. This *Profile* is a FullFeatured *Profile* that defines a minimum set of functionality required for generic OPC UA *Clients*. Such a *Client* shall be able to use local, subnet and global discovery. It shall be able to maintain a connection with a single *Session* (as required for nano embedded *Servers*). If *Subscriptions* are used, the *Client* shall respect the limits of *Servers* with limited resources. If a *Server* does not support *Subscriptions*, the *Client* shall provide read access as fallback. The *Client* shall provide connection establishment through the OPC UA TCP binary protocol with and without security.

This *Profile* supersedes the "Standard UA *Client Profile*".

Table 154 – Standard UA Client 2017 Profile

Group	Conformance Unit / *Profile* Title	Optional
Profile	*AddressSpace* Lookup *Client* Facet	False
Profile	*Attribute* Read *Client* Facet	False
Profile	*Attribute* Write *Client* Facet	False
Profile	Base *Client* Behaviour Facet	False
Profile	Core 2017 *Client* Facet	False
Profile	DataChange Subscriber *Client* Facet	False
Profile	*Discovery Client* Facet	False
Profile	Entry Level Support 2015 *Client* Facet	False
Profile	Global *Certificate* Management *Client* Facet	False
Profile	Global *Discovery Client* Facet	False
Profile	*Method Client* Facet	False
Profile	Subnet *Discovery Client* Facet	False
Profile	UA-TCP UA-SC UA-Binary	False
Profile	User Token – Anonymous Facet	False

6.6.143 UA-TCP UA-SC UA-Binary

Table 155 describes the details of the UA-TCP UA-SC UA-Binary. This transport Facet defines a combination of network protocol, security protocol and message encoding that is optimized for low resource consumption and high performance. It combines the simple TCP based network protocol UA-TCP 1.0 with the binary security protocol UA-SecureConversation 1.0 and the binary message encoding UA-Binary 1.0.

Table 155 – UA-TCP UA-SC UA-Binary

Group	Conformance Unit / *Profile* Title	Optional
Protocol and Encoding	Protocol UA TCP	False
Protocol and Encoding	UA Binary Encoding	False
Protocol and Encoding	UA Secure Conversation	False

6.6.144 HTTPS UA-Binary

Table 156 describes the details of the HTTPS UA-Binary. This transport Facet defines a combination of network protocol, security protocol and message encoding that balances compatibility with widely used HTTPS transport and a compact UA-Binary encoded message for added performance. It is expected that this transport will be used to support installations where firewalls only permit HTTPS or where a WEB browser is used as *Client*. This transport requires that one of the TransportSecurity *Profiles* for TLS be provided.

Table 156 – HTTPS UA-Binary

Group	Conformance Unit / *Profile* Title	Optional
Protocol and Encoding	Protocol HTTPS	False
Protocol and Encoding	UA Binary Encoding	False
Security	Security TLS General	False

6.6.145 HTTPS UA-XML

Table 157 describes the details of the HTTPS UA-XML. This transport Facet defines a combination of network protocol, security protocol and message encoding that uses HTTPS transport and a SOAP XML encoded message for use with standard SOAP V1.2 toolkits. This transport requires that one of the TransportSecurity *Profiles* for TLS be provided.

Table 157 – HTTPS UA-XML

Group	Conformance Unit / *Profile* Title	Optional
Protocol and Encoding	Protocol HTTPS	False
Protocol and Encoding	UA SOAP-XML Encoding	False
Security	Security TLS General	False

6.6.146 HTTPS UA-JSON

Table 158 describes the details of the HTTPS UA-JSON. This transport Facet defines a combination of network protocol, security protocol and message encoding that uses HTTPS transport and a UA-JSON encoded message. This transport requires that one of the TransportSecurity *Profiles* for TLS be provided.

Table 158 – HTTPS UA-JSON

Group	Conformance Unit / *Profile* Title	Optional
Protocol and Encoding	JSON Reversible Encoding	False
Protocol and Encoding	Protocol HTTPS	False
Security	Security TLS General	False

6.6.147 WSS UA-SC UA-Binary

Table 159 describes the details of the WSS UA-SC UA-Binary. This transport Facet defines a combination of network protocol, security protocol and message encoding that uses WSS transport as a tunnel for UA-SecureConversation and UA-Binary encoded messages. Although transport security is available in WSS via TLS, additional message security can be used to assure end-to-end security.

Table 159 – WSS UA-SC UA-Binary

Group	Conformance Unit / *Profile* Title	Optional
Protocol and Encoding	Protocol Web Sockets	False
Protocol and Encoding	UA Binary Encoding	False
Protocol and Encoding	UA Secure Conversation	False
Security	Security TLS General	False

6.6.148 WSS UA-JSON

Table 160 describes the details of the WSS UA-JSON. This transport Facet defines a combination of network protocol, security protocol and message encoding that uses WSS transport with UA-JSON encoded messages.

Table 160 – WSS UA-JSON

Group	Conformance Unit / *Profile* Title	Optional
Protocol and Encoding	JSON Reversible Encoding	False
Protocol and Encoding	Protocol Web Sockets	False
Security	Security TLS General	False

6.6.149 Security User Access Control Full

Table 161 describes the details of the Security User Access Control Full. A *Server* that supports this profile supports restricting multiple levels of access to all *Nodes* in the *AddressSpace* based on the validated user.

Table 161 – Security User Access Control Full

Group	Conformance Unit / *Profile* Title	Optional
Profile	Security User Access Control Base	False
Address Space Model	Address Space User Access Level Full	False

6.6.150 Security User Access Control Base

Table 162 describes the details of the Security User Access Control Base. A *Server* that supports this profile supports restricting some level of access to some *Nodes* in the *AddressSpace* based on the validated user.

Table 162 – Security User Access Control Base

Group	Conformance Unit / *Profile* Title	Optional
Address Space Model	Address Space User Access Level Base	False
Security	Security User IssuedToken Kerberos	True
Security	Security User IssuedToken Kerberos Windows	True
Security	Security User Name Password	False
Security	Security User X509	True

6.6.151 Security Time Synchronization

Table 163 describes the details of the Security Time Synchronization. This Facet indicates that the application supports the minimum required level of time synchronization to ensure secure communication. One of the optional time synchronization conformance units shall be supported.

Table 163 – Security Time Synchronization

Group	Conformance Unit / *Profile* Title	Optional
Security	Security Time Synch – Configuration	False
Security	Security Time Synch – NTP / OS Based support	True
Security	Security Time Synch – UA based support	True

6.6.152 Best Practice – Audit Events

Table 164 describes the details of the Best Practice – Audit Events. Subscriptions for Audit Events shall be restricted to authorized personnel.

Table 164 – Best Practice – Audit Events

Group	Conformance Unit / *Profile* Title	Optional
Miscellaneous	Best Practice – Audit Events	False

6.6.153 Best Practice – Alarm Handling

Table 165 describes the details of the Best Practice – *Alarm* Handling. A *Server* should restrict critical alarm handling functionality to users that have the appropriate rights to perform these actions.

Table 165 – Best Practice – Alarm Handling

Group	Conformance Unit / *Profile* Title	Optional
Miscellaneous	Best Practice – *Alarm* Handling	False

6.6.154 Best Practice – Random Numbers

Table 166 describes the details of the Best Practice – Random Numbers. All random numbers that are required for security should use appropriate cryptographic library based random number generators.

Table 166 – Best Practice – Random Numbers

Group	Conformance Unit / *Profile* Title	Optional
Miscellaneous	Best Practice – Random Numbers	False

6.6.155 Best Practice – Timeouts

Table 167 describes the details of the Best Practice – Timeouts. The administrator should be able to configure reasonable timeouts for Secure Channels, Sessions and Subscriptions. Setting these timeouts allows limiting Denial of *Service* attacks and overload issues.

Table 167 – Best Practice – Timeouts

Group	Conformance Unit / *Profile* Title	Optional
Miscellaneous	Best Practice – Timeouts	False

6.6.156 Best Practice – Administrative Access

Table 168 describes the details of the Best Practice – Administrative Access. The *Server* and *Client* allow restricting the use of certain *Services* and access to parts of the *AddressSpace* to administrative personnel. This includes multiple level of administrative access on platforms that support multiple administrative roles (such as Windows or Linux).

Table 168 – Best Practice – Administrative Access

Group	Conformance Unit / *Profile* Title	Optional
Miscellaneous	Best Practice – Administrative Access	False

6.6.157 Best Practice – Strict Message Handling

Table 169 describes the details of the Best Practice – Strict *Message* Handling. *Server* and *Client* reject messages that are incorrectly formed as specified in IEC 62541-4 and IEC 62541-6.

Table 169 – Best Practice – Strict Message Handling

Group	Conformance Unit / *Profile* Title	Optional
Miscellaneous	Best Practice – Strict *Message* Handling	False

6.6.158 Best Practice – Audit Events Client

Table 170 describes the details of the Best Practice – Audit Events *Client*. Audit Tracking system connect to a *Server* using a Secure Channel and under the appropriate authorization to allow access to Audit Events.

Table 170 – Best Practice – Audit Events Client

Group	Conformance Unit / *Profile* Title	Optional
Miscellaneous	Best Practice – Audit Events *Client*	False

6.6.159 TransportSecurity – TLS 1.2

Table 171 describes the details of the TransportSecurity – TLS 1.2. This Facet defines a transport security for configurations with high security needs. It makes use of TLS 1.2 and uses TLS_RSA_WITH_AES_256_CBC_SHA256. As computing power increases, security algorithms are expected to expire. NIST provides guidelines for expected expiration dates for individual algorithms. These guidelines provide recommended dates at which the algorithm should be replaced or upgraded to a more secure algorithm. They do not indicate a failure of the algorithm. NIST has no recommendations for this TransportSecurity. It is recommended that *Servers* and *Clients* support all security profiles and developers provide the recommended profile as a default. It is up to an administrator to configure the actual exposed TransportSecurity *Profiles*.

Table 171 – TransportSecurity – TLS 1.2

Group	Conformance Unit / *Profile* Title	Optional
Security	Security TLS_RSA with AES_256_CBC_SHA256	False

6.6.160 TransportSecurity – TLS 1.2 with PFS

Table 172 describes the details of the TransportSecurity – TLS 1.2 with PFS. This Facet defines a transport security for configurations with high security needs and perfect forward secrecy (PFS). It makes use of TLS 1.2 and uses TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 or TLS_DHE_RSA_WITH_AES_256_CBC_SHA256.

As computing power increases, security algorithms are expected to expire. NIST provides guidelines for expected expiration dates for individual algorithms. These guidelines provide recommended dates at which the algorithm should be replaced or upgraded to a more secure algorithm. They do not indicate a failure of the algorithm. NIST has no recommendations for this TransportSecurity. It is recommended that *Servers* and *Clients* support all security profiles and developers provide the recommended profile as a default. It is up to an administrator to configure the actual exposed TransportSecurity *Profiles*.

Table 172 – TransportSecurity – TLS 1.2 with PFS

Group	Conformance Unit / *Profile* Title	Optional
Security	Security TLS_DHE_RSA with AES_nnn_CBC_SHA256	False

6.6.161 SecurityPolicy – None

Table 173 describes the details of the SecurityPolicy – None. This security Facet defines a security policy used for configurations with the lowest security needs. This security policy can affect the behaviour of the CreateSession and *ActivateSession Services*. It also results in a SecureChannel which has no channel security. By default this security policy should be disabled if any other security policies are available.

Table 173 – SecurityPolicy – None

Group	Conformance Unit / *Profile* Title	Optional
Security	AsymmetricEncryptionAlgorithm_None	False
Security	AsymmetricSignatureAlgorithm_None	False
Security	KeyDerivationAlgorithm_None	False
Security	Security None CreateSession ActivateSession	False
Security	Security None CreateSession ActivateSession 1.0	True
Security	SecurtyPolicy_None_Limits	False
Security	SymmetricEncryptionAlgorithm_None	False
Security	SymmetricSignatureAlgorithm_None	False

6.6.162 SecurityPolicy – Basic128Rsa15

SecurityPolicy – Basic128Rsa15 has been deprecated in v1.04 since the hash algorithm Sha-1 is not considered secure anymore.

6.6.163 SecurityPolicy – Basic256

SecurityPolicy – Basic128Rsa15 has been deprecated in v1.04 since the hash algorithm Sha-1 is not considered secure anymore.

6.6.164 SecurityPolicy [A] – Aes128-Sha256-RsaOaep

Table 174 describes the details of the SecurityPolicy [A] – Aes128-Sha256-RsaOaep. This security Facet defines a security policy for configurations with average security needs. It requires a PKI infrastructure. As computing power increases, security policies are expected to expire. NIST provides guidelines for expected expiration dates for individual algorithms. These guidelines provide recommended dates at which the algorithm should be replaced or upgraded to a more secure algorithm. They do not indicate a failure of the algorithm. This security policy has no published end dates as of this time. It is recommended that *Servers* and *Clients* support all security profiles and support configurability of actual exposed and default security policies.

Table 174 – SecurityPolicy [A] – Aes128-Sha256-RsaOaep

Group	Conformance Unit / *Profile* Title	Optional
Security	Aes128-Sha256-RsaOaep_Limits	False
Security	AsymmetricEncryptionAlgorithm_RSA-OAEP-SHA1	False
Security	AsymmetricSignatureAlgorithm_RSA-PKCS15-SHA2-256	False
Security	CertificateSignatureAlgorithm_RSA-PKCS15-SHA2-256	False
Security	KeyDerivationAlgorithm_P-SHA2-256	False
Security	Security *Certificate* Validation	False
Security	Security Encryption Required	False
Security	Security Signing Required	False
Security	SymmetricEncryptionAlgorithm_AES128-CBC	False
Security	SymmetricSignatureAlgorithm_HMAC-SHA2-256	False

6.6.165 SecurityPolicy [B] – Basic256Sha256

Table 175 describes the details of the SecurityPolicy [B] – Basic256Sha256. This security Facet defines a security policy for configurations with high security needs. It requires a PKI infrastructure.

As computing power increases, security policies are expected to expire. NIST provides guidelines for expected expiration dates for individual algorithms. These guidelines provided recommended dates at which the algorithm should be replaced or upgraded to a more secure algorithm. They do not indicate a failure of the algorithm. This security policy has no published end dates as of this time. It is recommended that *Servers* and *Clients* support all security profiles and developers provide the recommended profile as a default. It is up to an administrator to configure the actual exposed security policies.

Table 175 – SecurityPolicy [B] – Basic256Sha256

Group	Conformance Unit / *Profile* Title	Optional
Security	AsymmetricEncryptionAlgorithm_RSA-OAEP-SHA1	False
Security	AsymmetricSignatureAlgorithm_RSA-PKCS15-SHA2-256	False
Security	Basic256Sha256_Limits	False
Security	CertificateSignatureAlgorithm_RSA-PKCS15-SHA2-256	False
Security	KeyDerivationAlgorithm_P-SHA2-256	False
Security	Security *Certificate* Validation	False
Security	Security Encryption Required	False
Security	Security Signing Required	False
Security	SymmetricEncryptionAlgorithm_AES256-CBC	False
Security	SymmetricSignatureAlgorithm_HMAC-SHA2-256	False

6.6.166 SecurityPolicy – Aes256-Sha256-RsaPss

Table 176 describes the details of the SecurityPolicy – Aes256-Sha256-RsaPss. This security Facet defines a security policy for configurations with a need for high security. It requires a PKI infrastructure. As computing power increases, security policies are expected to expire. NIST provides guidelines for expected expiration dates for individual algorithms. These guidelines provide recommended dates at which the algorithm should be replaced or upgraded to a more secure algorithm. They do not indicate a failure of the algorithm. This security policy has no published end dates as of this time. It is recommended that *Servers*

and *Clients* support all security profiles and support configurability of actual exposed and default security policies.

Table 176 – SecurityPolicy – Aes256-Sha256-RsaPss

Group	Conformance Unit / *Profile* Title	Optional
Security	Aes256-Sha256-RsaPss_Limits	False
Security	AsymmetricEncryptionAlgorithm_RSA-OAEP-SHA2-256	False
Security	AsymmetricSignatureAlgorithm_RSA-PSS -SHA2-256	False
Security	CertificateSignatureAlgorithm_ RSA-PKCS15-SHA2-256	False
Security	KeyDerivationAlgorithm_P-SHA2-256	False
Security	Security *Certificate* Validation	False
Security	Security Encryption Required	False
Security	Security Signing Required	False
Security	SymmetricEncryptionAlgorithm_AES256-CBC	False
Security	SymmetricSignatureAlgorithm_HMAC-SHA2-256	False

6.6.167 User Token – Anonymous Facet

Table 177 describes the details of the User Token – Anonymous Facet. This Facet indicates that anonymous User Tokens are supported.

Table 177 – User Token – Anonymous Facet

Group	Conformance Unit / *Profile* Title	Optional
Security	Security User Anonymous	False

6.6.168 User Token – User Name Password Server Facet

Table 178 describes the details of the User Token – User Name Password *Server* Facet. This Facet indicates that a user token that is comprised of a username and password is supported. This user token can affect the behaviour of the *ActivateSession Service*.

Table 178 – User Token – User Name Password Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Security	Security Invalid user token	False
Security	Security User Name Password	False

6.6.169 User Token – X509 Certificate Server Facet

Table 179 describes the details of the User Token – X509 *Certificate Server* Facet. This Facet indicates that the use of an X509 certificates to identify users is supported.

Table 179 – User Token – X509 Certificate Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Security	Security Invalid user token	False
Security	Security User X509	False

6.6.170 User Token – Issued Token Server Facet

Table 180 describes the details of the User Token – Issued Token *Server* Facet. This Facet indicates that a User Token that is comprised of an issued token is supported.

Table 180 – User Token – Issued Token Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Security	Security Invalid user token	False
Security	Security User IssuedToken Kerberos	False

6.6.171 User Token – Issued Token Windows Server Facet

Table 181 describes the details of the User Token – Issued Token Windows *Server* Facet. This Facet further refines the User Token – Issued Token to indicate a windows implementation of Kerberos

Table 181 – User Token – Issued Token Windows Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	User Token – Issued Token *Server* Facet	False
Security	Security User IssuedToken Kerberos Windows	False

6.6.172 User Token – JWT Server Facet

Table 182 describes the details of the User Token – JWT *Server* Facet. This Facet defines support for JSON Web Tokens (JWT) to identify the user during *Session* setup. A JWT is the Access Token format which OPC UA requires when using OAuth2.

Table 182 – User Token – JWT Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Security	Azure Identity Provider Authority *Profile*	True
Security	OAuth2 Authority *Profile*	True
Security	OPC UA Authority *Profile*	True
Security	Security Invalid user token	False
Security	Security User JWT IssuedToken	False
Security	Security User JWT Token Policy	False

6.6.173 User Token – User Name Password Client Facet

Table 183 describes the details of the User Token – User Name Password *Client* Facet. This Facet defines the ability to use a user token that is comprised of a username and password.

Table 183 – User Token – User Name Password Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Security	Security User Name Password *Client*	False

6.6.174 User Token – X509 Certificate Client Facet

Table 184 describes the details of the User Token – X509 *Certificate Client* Facet. This Facet defines the ability to use an X509 certificates to identify users.

Table 184 – User Token – X509 Certificate Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Security	Security User X509 *Client*	False

6.6.175 User Token – Issued Token Client Facet

Table 185 describes the details of the User Token – Issued Token *Client* Facet. This Facet defines the ability to use the User Token – Issued Token (Kerberos) to connect to a *Server*.

Table 185 – User Token – Issued Token Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Security	Security User IssuedToken Kerberos *Client*	False

6.6.176 User Token – Issued Token Windows Client Facet

Table 186 describes the details of the User Token – Issued Token Windows *Client* Facet. This Facet defines the ability to use the User Token – Issued Token (Windows implementation of Kerberos) to connect to a *Server*

Table 186 – User Token – Issued Token Windows Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Security	Security User IssuedToken Kerberos Windows *Client*	False

6.6.177 User Token – JWT Client Facet

Table 187 describes the details of the User Token – JWT *Client* Facet. This Facet defines the ability to use JSON Web Tokens (JWT) as user identification during *Session* setup. JWTs are used to request an access token from an external Authorization *Service*.

Table 187 – User Token – JWT Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Security	Azure Identity Provider Authority *Profile*	True
Security	OAuth2 Authority *Profile*	True
Security	OPC UA Authority *Profile*	True
Security	Security User JWT IssuedToken *Client*	False
Security	Security User JWT Token Policy *Client*	False

6.6.178 Global Discovery Server Profile

Global Discovery Server Profile has been deprecated as it is superseded with Global Discovery Server 2017 Profile in v1.04.

6.6.179 Global Discovery Server 2017 Profile

Table 188 describes the details of the Global *Discovery Server* 2017 *Profile*. This *Profile* is a FullFeatured *Profile* that covers the necessary *Services* and Information Model of a UA *Server* that acts as a GDS.

This *Profile* supersedes the "Global *Discovery Server Profile*".

Table 188 – Global Discovery Server 2017 Profile

Group	Conformance Unit / *Profile* Title	Optional
Profile	Core 2017 *Server* Facet	False
Profile	*Method Server* Facet	False
Profile	Standard DataChange *Subscription* 2017 *Server* Facet	False
Profile	UA-TCP UA-SC UA-Binary	False
GDS	GDS Application Directory	False
GDS	GDS LDS-ME Connectivity	False
GDS	GDS Query Applications	False
Security	Security Default ApplicationInstance *Certificate*	False
Security	Security Policy Required	False
Session Services	*Session* Minimum 50 Parallel	False

6.6.180 Global Discovery and Certificate Management Server

Global Discovery and Certificate Management Server Profile has been deprecated as it is superseded with Global Discovery and Certificate Mgmt 2017 Server Profile in v1.04.

6.6.181 Global Discovery and Certificate Mgmt 2017 Server

Table 189 describes the details of the Global *Discovery* and *Certificate* Mgmt 2017 *Server*. This *Profile* is a FullFeatured *Profile* that covers the necessary *Services* and Information Model of a UA *Server* that acts as a GDS and a global *Certificate* Manager.

This *Profile* supersedes the "Global *Discovery* and *Certificate* Management *Server*".

Table 189 – Global Discovery and Certificate Mgmt 2017 Server

Group	Conformance Unit / *Profile* Title	Optional
Profile	Auditing *Server* Facet	False
Profile	File Access *Server* Facet	False
Profile	Global *Discovery Server* 2017 *Profile*	False
Profile	Standard *Event Subscription Server* Facet	False
GDS	GDS *Certificate* Manager Pull Model	False

6.6.182 Global Certificate Management Client Profile

Global Certificate Management Client Profile has been deprecated as it is superseded with Global Certificate Management Client 2017 Profile in v1.04.

6.6.183 Global Certificate Management Client 2017 Profile

Table 190 describes the details of the Global *Certificate* Management *Client* 2017 *Profile*. This *Profile* is a FullFeatured *Profile* that uses the Push Model for the management of *Certificates* and Trust Lists.

This *Profile* supersedes the "Global *Certificate* Management *Client Profile*".

Table 190 – Global Certificate Management Client 2017 Profile

Group	Conformance Unit / *Profile* Title	Optional
Profile	Core 2017 *Client* Facet	False
Profile	*Discovery Client* Facet	False
Profile	Entry Level Support 2015 *Client* Facet	False
Profile	File Access *Client* Facet	False
Profile	*Method Client* Facet	False
Profile	UA-TCP UA-SC UA-Binary	False
GDS	GDS *Certificate* Manager Push Model	False
Security	Security Default ApplicationInstance *Certificate*	False

6.6.184 Global Service Authorization Request Server Facet

Table 191 describes the details of the Global *Service* Authorization Request *Server* Facet. This Facet defines the capability of a *Server* (like a GDS) to provide access tokes to OPC UA *Clients* via an Authorization *Service* as defined in IEC 62541-12.

Table 191 – Global Service Authorization Request Server Facet

Group	Conformance Unit / *Profile* Title	Optional
GDS	GDS Authorization *Service Server*	False

6.6.185 Global Service KeyCredential Pull Facet

Table 192 describes the details of the Global Service KeyCredential Pull Facet. This Facet requires providing the *Information Model* for Pull Management as defined in IEC 62541-12. For example KeyCredentials are needed to access an Authorization *Service* or a Broker. OPC UA *Clients* use this Information Model to request and update KeyCredentials they need.

Table 192 – Global Service KeyCredential Pull Facet

Group	Conformance Unit / *Profile* Title	Optional
GDS	GDS Key Credential *Service* Pull Model	False

6.6.186 Global Service KeyCredential Push Facet

Table 193 describes the details of the Global Service KeyCredential Push Facet. This Facet requires the use of KeyCredential Push Management functions to set or update credentials in an OPC UA Server. For example KeyCredentials are needed to access an Authorization *Service* or a Broker. This OPC UA *Server* in turn has to provide the KeyCredentialConfigurationType *Objects* that represent required credentials.

Table 193 – Global Service KeyCredential Push Facet

Group	Conformance Unit / *Profile* Title	Optional
GDS	GDS Key Credential *Service* Push Model	False

Bibliography

Test Specifications

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*

SOMMAIRE

COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

_____

ARCHITECTURE UNIFIÉE OPC –

Partie 7: Profils

AVANT-PROPOS

1) La Commission Electrotechnique Internationale (IEC) est une organisation mondiale de normalisation composée de l'ensemble des comités électrotechniques nationaux (Comités nationaux de l'IEC). L'IEC a pour objet de favoriser la coopération internationale pour toutes les questions de normalisation dans les domaines de l'électricité et de l'électronique. A cet effet, l'IEC – entre autres activités – publie des Normes internationales, des Spécifications techniques, des Rapports techniques, des Spécifications accessibles au public (PAS) et des Guides (ci-après dénommés "Publication(s) de l'IEC"). Leur élaboration est confiée à des comités d'études, aux travaux desquels tout Comité national intéressé par le sujet traité peut participer. Les organisations internationales, gouvernementales et non gouvernementales, en liaison avec l'IEC, participent également aux travaux. L'IEC collabore étroitement avec l'Organisation Internationale de Normalisation (ISO), selon des conditions fixées par accord entre les deux organisations.
2) Les décisions ou accords officiels de l'IEC concernant les questions techniques représentent, dans la mesure du possible, un accord international sur les sujets étudiés, étant donné que les Comités nationaux de l'IEC intéressés sont représentés dans chaque comité d'études.
3) Les Publications de l'IEC se présentent sous la forme de recommandations internationales et sont agréées comme telles par les Comités nationaux de l'IEC. Tous les efforts raisonnables sont entrepris afin que l'IEC s'assure de l'exactitude du contenu technique de ses publications; l'IEC ne peut pas être tenue responsable de l'éventuelle mauvaise utilisation ou interprétation qui en est faite par un quelconque utilisateur final.
4) Dans le but d'encourager l'uniformité internationale, les Comités nationaux de l'IEC s'engagent, dans toute la mesure possible, à appliquer de façon transparente les Publications de l'IEC dans leurs publications nationales et régionales. Toutes divergences entre toutes Publications de l'IEC et toutes publications nationales ou régionales correspondantes doivent être indiquées en termes clairs dans ces dernières.
5) L'IEC elle-même ne fournit aucune attestation de conformité. Des organismes de certification indépendants fournissent des services d'évaluation de conformité et, dans certains secteurs, accèdent aux marques de conformité de l'IEC. L'IEC n'est responsable d'aucun des services effectués par les organismes de certification indépendants.
6) Tous les utilisateurs doivent s'assurer qu'ils sont en possession de la dernière édition de cette publication.
7) Aucune responsabilité ne doit être imputée à l'IEC, à ses administrateurs, employés, auxiliaires ou mandataires, y compris ses experts particuliers et les membres de ses comités d'études et des Comités nationaux de l'IEC, pour tout préjudice causé en cas de dommages corporels et matériels, ou de tout autre dommage de quelque nature que ce soit, directe ou indirecte, ou pour supporter les coûts (y compris les frais de justice) et les dépenses découlant de la publication ou de l'utilisation de cette Publication de l'IEC ou de toute autre Publication de l'IEC, ou au crédit qui lui est accordé.
8) L'attention est attirée sur les références normatives citées dans cette publication. L'utilisation de publications référencées est obligatoire pour une application correcte de la présente publication.
9) L'attention est attirée sur le fait que certains des éléments de la présente Publication de l'IEC peuvent faire l'objet de droits de brevet. L'IEC ne saurait être tenue pour responsable de ne pas avoir identifié de tels droits de brevets et de ne pas avoir signalé leur existence.

La Norme internationale IEC 62541-7 a été établie par le sous-comité 65E: Les dispositifs et leur intégration dans les systèmes de l'entreprise, du comité d'études 65 de l'IEC: Mesure, commande et automation dans les processus industriels.

Cette troisième édition annule et remplace la deuxième édition parue en 2015. Cette édition constitue une révision technique.

Cette édition inclut les modifications techniques majeures suivantes par rapport à l'édition précédente:

a) ajout de nouveaux profils fonctionnels:
   - profils pour la découverte globale et la gestion globale des certificats;
   - profils pour la gestion globale de KeyCredential et pour la gestion globale des jetons d'accès;

- facette pour les abonnements durables;
- profil client UA normalisé;
- profils pour l'administration des rôles et permissions des utilisateurs.

b) ajout de nouveaux profils de transport:
- HTTPS avec codage JSON;
- Secure WebSockets (WSS) avec codage binaire ou JSON;
- connectivité inversée.

c) ajout de nouveaux profils de sécurité:
- TransportSecurity – TLS 1.2 avec PFS (confidentialité persistante);
- SecurityPolicy [A] – Aes128-Sha256-RsaOaep (en remplacement de Base128Rsa15);
- SecurityPolicy – Aes256-Sha256-RsaPss ajoute la confidentialité persistante pour UA TCP);
- jeton d'utilisateur JWT (Jason Web Token).

d) spécification des profils de sécurité déconseillés (en raison des algorithmes cassés):
- SecurityPolicy – Basic128Rsa15 (algorithme Sha1 cassé);
- SecurityPolicy – Basic256 (algorithme Sha1 cassé);
- TransportSecurity – TLS 1.0 (algorithme RC4 cassé);
- TransportSecurity – TLS 1.1 (algorithme RC4 cassé).

e) transport déconseillé (absence de prise en charge sur la plupart des plateformes):
- SOAP/HTTP avec WS-SecureConversation (tous les codages).

Le texte de cette Norme internationale est issu des documents suivants:

FDIS	Rapport de vote
65E/707/FDIS	65E/725/RVD

Le rapport de vote indiqué dans le tableau ci-dessus donne toute information sur le vote ayant abouti à l'approbation de cette Norme internationale.

Ce document a été rédigé selon les Directives ISO/IEC, Partie 2.

Dans l'ensemble du présent document et dans les autres parties de la série IEC 62541, certaines conventions de document sont utilisées:

Le format *italique* est utilisé pour mettre en évidence un terme défini ou une définition qui apparaît à l'article "Termes et définitions" dans l'une des parties de la série IEC 62541.

Le format *italique* est également utilisé pour mettre en évidence le nom d'un paramètre d'entrée ou de sortie de service, ou le nom d'une structure ou d'un élément de structure habituellement défini dans les tableaux.

Par ailleurs, les *termes* et les *noms en italique* sont, à quelques exceptions près, écrits en camel-case (pratique qui consiste à joindre, sans espace, les éléments des mots ou expressions composés, la première lettre de chaque élément étant en majuscule). Par exemple, le terme défini est *AddressSpace* et non Espace d'adressage. Cela permet de mieux comprendre qu'il existe une définition unique pour *AddressSpace*, et non deux définitions distinctes pour Espace et pour Adressage.

Une liste de toutes les parties de la série IEC 62541, publiées sous le titre général *Architecture unifiée OPC*, peut être consultée sur le site web de l'IEC.

Le comité a décidé que le contenu de ce document ne sera pas modifié avant la date de stabilité indiquée sur le site web de l'IEC sous "http://webstore.iec.ch" dans les données relatives au document recherché. A cette date, le document sera

- reconduit,
- supprimé,
- remplacé par une édition révisée, ou
- amendé.

IMPORTANT – Le logo *"colour inside"* qui se trouve sur la page de couverture de cette publication indique qu'elle contient des couleurs qui sont considérées comme utiles à une bonne compréhension de son contenu. Les utilisateurs devraient, par conséquent, imprimer cette publication en utilisant une imprimante couleur.

ARCHITECTURE UNIFIÉE OPC –

Partie 7: Profils

1 Domaine d'application

La présente partie de l'IEC 62541 définit les *Profils* de l'architecture unifiée OPC (OPC UA). Les *Profils* du présent document permettent de classer les caractéristiques en fonction des essais de produits OPC UA et de la nature des essais (via un outil ou en laboratoire). Cela inclut les essais effectués à l'aide de l'outil d'essai de conformité CTT OPC UA développé par la Fondation OPC (outil d'essai autonome), ainsi que les essais réalisés par des laboratoires de certification indépendants de cette même fondation. Le présent document peut également faire référence aux outils d'essai ou au laboratoire d'essai d'un autre organisme. Dans le cas présent, l'élément important est le concept qui oppose les essais fondés sur un outil automatisé et les essais en laboratoire. Le domaine d'application de la présente norme inclut la définition d'une fonctionnalité qui ne peut être soumise à l'essai qu'en laboratoire, ainsi que la définition du regroupement des fonctionnalités à utiliser durant les essais de produits OPC UA effectués en laboratoire ou à l'aide d'outils automatisés. La définition des *TestCases* réels ne relève pas du domaine d'application du présent document, mais les catégories de *TestCases* générales relèvent du domaine d'application du présent document.

La plupart des applications OPC UA sont conformes à plusieurs *Profils*, mais pas à l'ensemble des *Profils*.

2 Références normatives

Les documents ci-après sont des références normatives indispensables à l'application du présent document. Pour les références datées, seule l'édition citée s'applique. Pour les références non datées, la dernière édition du document de référence s'applique (y compris les éventuels amendements).

IEC TR 62541-1, *OPC Unified Architecture – Part 1: Overview and Concepts* (disponible en anglais seulement)

IEC TR 62541-2, *OPC Unified Architecture – Part 2: Security Model* (disponible en anglais seulement)

IEC 62541-3, *Architecture unifiée OPC – Partie 3: Modèle d'espace d'adressage*

IEC 62541-4, *Architecture unifiée OPC – Partie 4: Services*

IEC 62541-5, *Architecture unifiée OPC – Partie 5: Modèle d'information*

IEC 62541-6, *Architecture unifiée OPC – Partie 6: Mappings*

IEC 62541-8, *Architecture unifiée OPC – Partie 8: Accès aux données*

IEC 62541-9, *Architecture unifiée OPC – Partie 9: Alarmes et conditions*

IEC 62541-11, *Architecture unifiée OPC – Partie 11: Accès à l'historique*

IEC 62541-12, *Architecture unifiée OPC – Partie 12: Services globaux et de découverte*

IEC 62541-13, *Architecture unifiée OPC – Partie 13: Agrégats*

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 Termes, définitions et termes abrégés

3.1 Termes et définitions

Pour les besoins du présent document, les termes et définitions donnés dans l'IEC TR 62541-1, l'IEC TR 62541-2, l'IEC 62541-3, l'IEC 62541-4, l'IEC 62541-6 et l'IEC 62541-8 ainsi que les suivants s'appliquent.

L'ISO et l'IEC tiennent à jour des bases de données terminologiques destinées à être utilisées en normalisation, consultables aux adresses suivantes:

- IEC Electropedia: disponible à l'adresse http://www.electropedia.org/
- ISO Online browsing platform: disponible à l'adresse http://www.iso.org/obp

NOTE La Figure 1 fournit une vue d'ensemble des termes définis dans le présent document et de leurs interactions.

3.1.1
application
programme logiciel qui exécute ou met en œuvre certains aspects d'OPC UA

Note 1 à l'article: L'application peut fonctionner sur tout type de machine et exécuter toute fonction, de même qu'elle peut être logicielle ou matérielle. La seule exigence est qu'elle mette en œuvre OPC UA.

3.1.2
ConformanceUnit
ensemble spécifique de caractéristiques OPC UA qui peuvent être soumises à l'essai comme une entité individuelle

Note 1 à l'article: Une ConformanceUnit peut représenter un groupe de services, des portions de services ou des modèles d'information.

3.1.3
ConformanceGroup
groupe de ConformanceUnits auquel un nom est attribué

Note 1 à l'article: Ce regroupement est destiné uniquement à faciliter l'organisation des ConformanceUnits. Les ConformanceGroups types incluent les groupes propres à chaque jeu de services OPC UA et à chacune des normes du Modèle d'information.

3.1.4
Facette
profil dédié à une caractéristique spécifique qu'un Serveur ou Client peut exiger

Note 1 à l'article: Les Facettes sont généralement combinées pour former des Profils d'un niveau plus élevé. L'utilisation du terme Facette dans l'intitulé d'un Profil indique que le Profil donné n'est pas un Profil autonome.

3.1.5
Profil FullFeatured
Profil qui définit toutes les caractéristiques nécessaires à la construction d'une Application OPC UA fonctionnelle

Note 1 à l'article: Un Profil complet en particulier définit les exigences de transport et de sécurité.

3.1.6
ProfileCategory
catégorie qui organise les Profils en classes d'applications, tels que Serveur ou Client

Note 1 à l'article: Ces catégories permettent de déterminer le type d'Application pour lequel un Profil donné serait utilisé. Pour plus d'informations, voir 4.4.

3.1.7
TestCase
description technique d'un ensemble d'étapes nécessaires pour soumettre à l'essai une fonction ou un modèle d'information particulier

Note 1 à l'article: Les TestCases fournissent les informations suffisantes pour permettre à un développeur de les mettre en œuvre dans un code. Les TestCases fournissent également un récapitulatif complet des résultats d'exécution du code mis en œuvre attendus, ainsi que les éventuelles conditions préalables qu'il importe d'établir avant de pouvoir exécuter le TestCase.

3.1.8
TestLab
installation conçue pour fournir des services d'essai

Note 1 à l'article: Ces services incluent, entre autres, le personnel qui effectue directement les essais, les essais automatisés et un processus reproductible formel. La Fondation OPC a élaboré une norme complète décrivant les TestLabs OPC UA ainsi que les essais à organiser (voir Compliance Part 8 UA Server, Compliance Part 9 UA Client).

3.2 Termes abrégés

DA data access (accès aux données)

HA historical access (accès à l'historique)

IHM interface homme-machine

NIST National Institute of Standard and Technology (Institut national des normes et de la technologie)

PKI Public Key Infrastructure (Infrastructure à clés publiques)

RSA Rivest-Shamir-Adleman

UA Unified Architecture (Architecture unifiée)

4 Vue d'ensemble

4.1 Généralités

La série IEC 62541, qui est la norme d'architecture unifiée OPC, décrit un certain nombre de *Services* et différents modèles d'information. Ces *Services* et modèles d'information peuvent être désignés comme les caractéristiques d'un *Serveur* ou d'un *Client*. Il est nécessaire que les *Serveurs* et *Clients* puissent décrire les caractéristiques qu'ils prennent en charge et celles qu'ils souhaitent voir certifiées. Le présent document procède au regroupement de ces caractéristiques. Les caractéristiques individuelles sont regroupées en *ConformanceUnits* qui sont regroupées en *Profils*. La Figure 1 représente les interactions entre les *Profils*, les *ConformanceUnits* et les *TestCases*. Les flèches larges désignent les composants utilisés pour composer le parent. Par exemple, un *Profil* est construit à partir de *Profils* et de *ConformanceUnits*. La figure représente également une caractéristique de l'outil d'essai de conformité CTT (Compliance Test Tool) OPC UA, dans le sens où elle permet de vérifier par essai si un *Profil* invoqué satisfait à l'ensemble des *ConformanceUnits*. L'outil permet également de vérifier par essai toutes les autres *ConformanceUnits* et de consigner dans un rapport les *Profils* qui satisfont aux essais de conformité. Chaque *TestCases* est défini dans un document distinct (voir Compliance Part 8 UA Server et Compliance Part 9 UA Client). Les *TestCases* sont à leur retour associés aux *ConformanceUnits* adéquates définies dans le présent document. Cette relation est également reflétée par l'outil d'essai de conformité OPC UA.

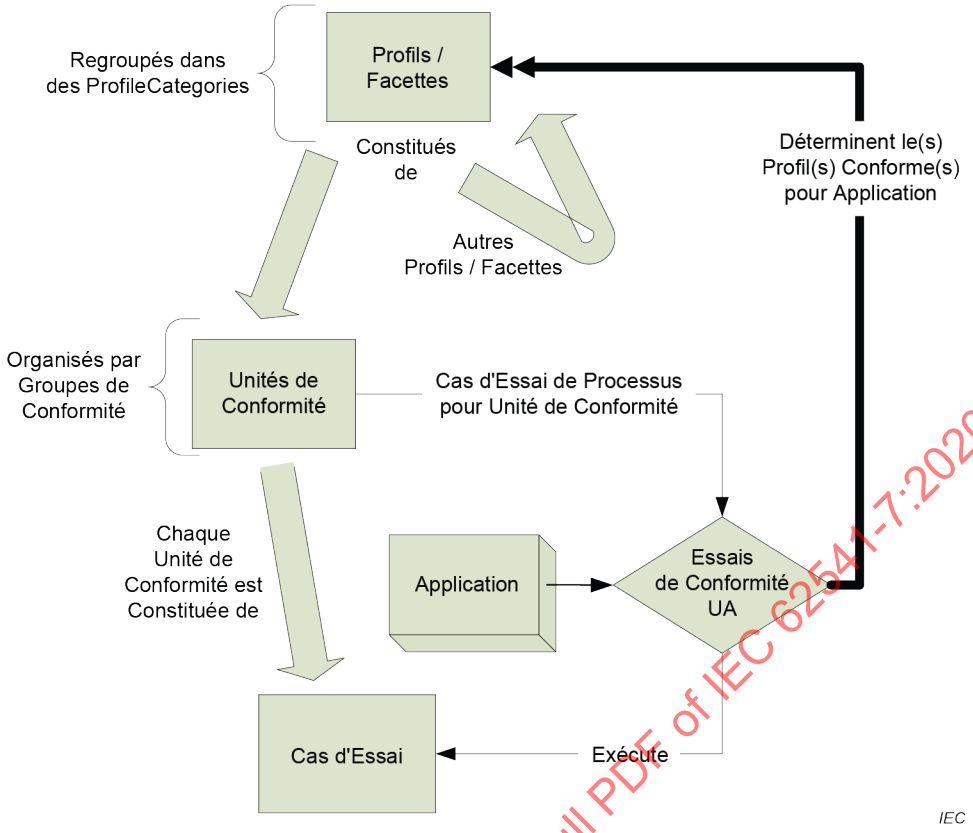


Figure 1 – Profil – ConformanceUnits – TestCases

4.2 ConformanceUnit

Chaque *ConformanceUnit* représente un ensemble spécifique de caractéristiques (un groupe de services, des portions de services ou des modèles d'information, par exemple) qui peuvent être soumises à l'essai en tant qu'entité individuelle. Les *ConformanceUnits* sont les blocs fonctionnels d'un *Profil*. Chaque *ConformanceUnit* peut également être utilisée comme catégorie d'essai. Pour chaque *ConformanceUnit,* il existe un certain nombre de TestCases qui permettent de soumettre à l'essai la fonctionnalité décrite par la *ConformanceUnit.* La description d'une *ConformanceUnit* est destinée à fournir des informations suffisantes pour représenter la fonctionnalité exigée. Dans de nombreux cas, il peut néanmoins être nécessaire que le lecteur examine également la partie applicable de l'OPC UA afin d'acquérir une compréhension complète de la *ConformanceUnit*. Des informations supplémentaires concernant les essais de la *ConformanceUnit* sont fournies dans les normes d'essai Compliance Part 8 UA Server ou Compliance Part 9 UA Client.

Les mêmes caractéristiques n'apparaissent pas dans plusieurs *ConformanceUnits*.

4.3 Profils

Un *Profil* est une agrégation nommée de *ConformanceUnits* et d'autres *Profils*. Pour prendre en charge un *Profil*, une application doit prendre en charge les *ConformanceUnits* et tous les *Profils* agrégés. La définition des *Profils* est une activité continuelle, dans le sens où il est prévu d'ajouter de nouveaux *Profils* ultérieurement.

Une application OPC UA prend généralement en charge plusieurs *Profils*.

Ces *Profils* peuvent comporter la même *ConformanceUnit*.

Lorsqu'un *Profil* est soumis à l'essai, ce sont les *ConformanceUnits* individuelles qui comportent le *Profil* qui sont soumises à l'essai.

Les *Profils* sont nommés selon les conventions de nommage (voir 6.3).

4.4 Catégories de profils

Les *Profils* sont regroupés en catégories afin d'aider les fournisseurs et utilisateurs finaux à comprendre le domaine d'application d'un *Profil*. Un *Profil* peut être attribué à une ou plusieurs catégories.

Le Tableau 1 contient la liste des *ProfileCategories* actuellement définies.

Tableau 1 – Catégories de profils

Catégorie	Description
Client	Les *Profils* de cette catégorie spécifient les fonctions d'un *Client* OPC UA.
Service d'annuaire global	Les *Profils* de cette catégorie spécifient les fonctions pour la découverte globale et la gestion des certificats.
Sécurité	Les *Profils* de cette catégorie spécifient les fonctions relatives à la Sécurité. Les politiques de sécurité font partie de cette catégorie. L'URI de ces politiques de sécurité doit faire partie intégrante d'une Description de point d'extrémité transmise par le service GetEndpoint. Les *Profils* de cette catégorie s'appliquent aux *Clients* et aux *Serveurs*.
Serveur	Les *Profils* de cette catégorie spécifient des fonctions pour un *Serveur* OPC UA. L'URI de ces *Profils* peut être présentée dans les fonctions du *Serveur*.
Transport	Les *Profils* de cette catégorie spécifient des mappings de protocoles spécifiques. L'URI de ce type de *Profils* doit faire partie intégrante d'une Description de point d'extrémité. Ces *Profils* s'appliquent aux *Clients* et aux *Serveurs*.

5 Unités de Conformité

5.1 Vue d'ensemble

Une *ConformanceUnit* représente une entité qui peut être soumise individuellement à l'essai. Pour faciliter la compréhension, la longue liste des *ConformanceUnits* a été regroupée en *ConformanceGroups*. Ces groupes reflètent les *Jeux de services* définis dans l'IEC 62541-4 et les modèles d'information OPC UA. Le Tableau 2 fournit la liste des *ConformanceGroups*. Ces groupes et les *ConformanceUnits* qu'ils décrivent sont décrits à l'Article 5, à partir du 5.2. Les *ConformanceGroups* n'ont aucune influence sur les essais. Ils sont uniquement utilisés pour des raisons d'organisation, autrement dit pour simplifier la lisibilité du présent document.

Tableau 2 – Groupes de Conformité

Groupe	Description
Modèle d'espace d'adressage	Définit les *ConformanceUnits* pour différentes caractéristiques de l'*AddressSpace* OPC UA.
Agrégats	Ensemble des *ConformanceUnits* relatives aux Agrégats, y compris les *ConformanceUnits* individuelles pour chaque Agrégat pris en charge, selon la description de l'IEC 62541-13.
Alarmes et *conditions*	Ensemble des *ConformanceUnits* associées au modèle d'information OPC UA pour les *Conditions*, *Conditions* acceptables, confirmations et *Alarmes* spécifiées dans l'IEC 62541-9.
Services Attribut	Inclut les *ConformanceUnits* permettant de lire ou d'écrire les valeurs d'*Attribut* actuelles ou historiques.
Audits	La sécurité au niveau utilisateur inclut la prise en charge des journaux d'audit de sécurité, avec une traçabilité entre les listes de contrôle *Client* et *Serveur*.
Informations de base	Tous les éléments d'information définis dans l'IEC 62541-5.
Accès aux données	*ConformanceUnits* spécifiques aux *Clients* et aux *Serveurs* qui traitent de la représentation et de l'utilisation des données d'automatisation, selon les spécifications de l'IEC 62541-8.
Services Découverte	*ConformanceUnits* axées sur la *Découverte* des Points d'extrémité du *Serveur*.
GDS	*ConformanceUnits* d'un GDS. Comprend des unités pour la découverte globale et la gestion globale des certificats.
Accès à l'historique	Accès aux données archivées des valeurs d'*Attribut de Nœud* ou Evénements.
Services Méthode	Les Méthodes représentent les appels de fonction des *Objets*. Les Méthodes sont invoquées et renvoyées uniquement après exécution (réussite ou échec).
Divers	Ce groupe contient des *ConformanceUnits* qui couvrent des sujets divers, tels que les comportements recommandés, la documentation, etc. Généralement, ces *ConformanceUnits* ne correspondent à aucun des autres groupes.
Services Eléments surveillés	Les *Clients* définissent des *MonitoredItems* pour s'abonner à des données et à des Evénements. Chaque *MonitoredItem* identifie l'élément à surveiller et l'*Abonnement* à utiliser pour envoyer des *Notifications*.
Services Gestion des *Nœuds*	Groupe les *ConformanceUnits* pour tous les *Services* afin d'ajouter et de supprimer des *Nœuds* et des *Références* de l'*AddressSpace* OPC UA.
Protocole et Codage	Couvre toutes les combinaisons de transport et de codage spécifiées dans l'IEC 62541-6.
Redondance	La conception d'OPC UA permet aux fournisseurs de créer des *Clients* et des *Serveurs* redondants de manière cohérente. La Redondance peut être utilisée pour une disponibilité élevée, la tolérance aux anomalies et l'équilibrage des charges.
Sécurité	*ConformanceUnits* relatives à la sécurité qui peuvent être intégrées dans des profils (cela couvre tous les aspects relatifs à la sécurité).
Services Session	Une *Session* (OPC UA) est une connexion de couche d'application.
Services Abonnement	Les abonnements permettent de reporter les *Notifications* au *Client*.
Services Vue	Les *Clients* utilisent le *Jeu de services* Vue pour naviguer dans l'*AddressSpace* OPC UA ou dans une vue (un sous-ensemble) de l'*AddressSpace* OPC UA.

5.2 Services

Les Tableaux 3 à 10 décrivent les *ConformanceUnits* pour les *Services* spécifiés dans l'IEC 62541-4. Les tableaux sont associés aux *Jeux de services.*

Une *ConformanceUnit* simple peut référencer plusieurs *Services* (CreateSession, ActivateSession et CloseSession, par exemple), mais elle peut également faire référence aux aspects individuels des *Services* (l'utilisation d'ActivateSession pour masquer un nouvel utilisateur, par exemple).

Chaque tableau fournit une liste des *Catégories de profils* auxquels appartient une *ConformanceUnit*, ainsi que le titre et la description de la *ConformanceUnit.* Dans certains cas, une *ConformanceUnit* est dérivée d'une autre *ConformanceUnit*. Cette unité parente sera donc spécifiée dans la description de chacune des unités dérivées. Dans ce cas, les unités dérivées héritent de tous les essais de leurs parents, avec un ou plusieurs TestCases supplémentaires. Ces TestCases peuvent seulement restreindre les TestCases existants. C'est le cas par exemple d'un TestCase visant à soumettre à l'essai le nombre de connexions, où le TestCase du parent exige au moins une connexion et où la *ConformanceUnit* obtenue exige un *TestCase* pour au moins cinq connexions.

Le *Jeu de services "Découverte"* comprend plusieurs *ConformanceUnits* (voir Tableau 3). Tous les *Serveurs* fournissent certains aspects de cette fonctionnalité; se reporter aux *Profils* catégorisés comme *Profils de serveur* pour plus d'informations. Les *Clients* peuvent prendre en charge certains aspects de cette fonctionnalité; se reporter aux *Profils* catégorisés comme *Profils de client* pour plus d'informations.

Tableau 3 – Services Découverte

Catégorie	Intitulé	Description
Serveur	*Découverte* "Obtenir des Points d'extrémité"	Prend en charge le *Service* GetEndpoints afin d'obtenir tous les points d'extrémité du *Serveur*. Cela inclut le filtrage fondé sur les *Profils*.
Serveur	*Découverte* "Obtenir les points d'extrémité SessionLess"	Prend en charge au moins un point d'extrémité pour émettre les Services SessionLess. Prend en charge l'obtention de ces points d'extrémité en acceptant l'URI de Transport comme filtre pour le Service GetEndpoints avec la chaîne de requête "SL" annexée à l'URI de Transport. Par exemple, "http://opcfoundation.org/UA-Profile/Transport/https-uajson?SL"
Serveur	*Découverte* "Trouver des *Serveurs* individuellement"	Prend en charge le *Service* FindServers uniquement pour un usage individuel.
Serveur	*Découverte* "Register"	Appelle le *Service* RegisterServer pour s'enregistrer (*Serveur* OPC UA) auprès d'un *Service "Découverte"* externe via un canal sécurisé par un SecurityMode autre que NONE.
Serveur	*Découverte* "Register2"	Appelle le *Service* RegisterServer2 pour s'enregistrer auprès d'un *Service "Découverte"* externe via un Canal sécurisé par un SecurityMode autre que "None". Cela comprend le passage d'une liste de brefs identificateurs de capacité. Les identificateurs et leur utilisation sont spécifiés dans l'IEC 62541-12.
Serveur	Notification d'un *Serveur "Découverte"* au moyen du mDNS	Fournit une fonction mDNS pour notifier un *Serveur* avec ses capacités. Les identificateurs de capacité et l'utilisation des journaux mDNS aux fins de la *Découverte* OPC UA sont spécifiés dans l'IEC 62541-12. Noter que cette fonctionnalité est exigée uniquement pour les *Serveurs* qui ne s'enregistrent pas avec un LDS. Les identificateurs de capacité et leur utilisation dans les journaux mDNS sont spécifiés dans l'IEC 62541-12.
Serveur	*Découverte* "Configuration"	Permet la configuration de l'URL du *Serveur "Découverte"*, où le *Serveur* s'enregistre lui-même. Permet la désactivation complète de l'enregistrement avec un *Serveur "Découverte"*.
Client	*Découverte Client* "Trouver les *Serveurs* de base"	Utilise le *Service* FindServers pour obtenir tous les *Serveurs* installés sur une plateforme donnée.
Client	*Découverte Client* "Trouver les *Serveurs* ayant un URI"	Utilise le *Service* FindServers pour obtenir les URL des URI des *Serveurs* spécifiques.

Catégorie	Intitulé	Description
Client	*Découverte Client* "Trouver les *Serveurs* dynamiques"	Détecte les nouveaux *Serveurs* après un appel initial du *Service* FindServers.
Client	*Découverte Client* "Trouver les *Serveurs* sur le Réseau"	Prend en charge l'une des options permettant de localiser les *Serveurs* dans le réseau.
Client	*Découverte Client* "Trouver les *Serveurs* sur le Réseau au moyen d'un LDS-ME"	Utilise le *Service* FindServersOnNetwork pour obtenir les URL des URI des *Serveurs* spécifiques. Noter que ce *Service* est disponible au moyen du *Serveur "Découverte"* Local avec l'extension de multidiffusion (LDS-ME).
Client	*Découverte Client* "Trouver les *Serveurs* sur le Réseau au moyen du mDNS"	Utilise la *Découverte* de *Service* au moyen du mDNS pour localiser les *Serveurs* d'un même réseau de multidiffusion. Le contenu des journaux mDNS pour la *Découverte* OPC UA est décrit dans l'IEC 62541-12. Noter que cette fonctionnalité est exigée pour les *Clients* uniquement lorsqu'il n'y a pas de *Serveur "Découverte"* Local comportant l'extension de multidiffusion (LDS-ME). Les identificateurs de capacité et leur utilisation dans les journaux mDNS sont spécifiés dans l'IEC 62541-12.
Client	*Découverte Client* "Trouver les *Serveurs* dans le GDS"	Utilise la *Méthode* QueryServers sur l'*Objet* du répertoire du GDS pour localiser les *Serveurs* qui remplissent les critères du filtre spécifié dans la requête. Cette *Méthode* est spécifiée dans l'IEC 62541-12.
Client	*Découverte Client* "Trouver les Applications dans le GDS"	Utilise la *Méthode* QueryApplications sur l'*Objet* du Répertoire du GDS pour localiser les *Applications* qui remplissent les critères du filtre spécifié dans la requête. Cette *Méthode* est spécifiée dans l'IEC 62541-12.
Client	*Découverte Client* "Obtenir les Points d'extrémité de base"	Utilise le *Service* GetEndpoints pour obtenir tous les Points d'extrémité d'un URI de *Serveur* donné.
Client	*Découverte Client* "Obtenir les Points d'extrémité SessionLess"	Utilise le *Service* GetEndpoints avec un filtre pour obtenir les Points d'extrémité qui peuvent être utilisés pour l'invocation du *Service* SessionLess. Le filtre est l'URI de Transport étendue par la chaîne de requête "SL". Par exemple, http://opcfoundation.org/UA-*Profile*/Transport/https-uajson?SL.
Client	*Découverte Client* "Obtenir les Points d'extrémité dynamiques"	Détecte les modifications apportées sur les Points d'extrémité après un appel initial du *Service* GetEndpoints.
Client	*Découverte Client* "Configurer les Points d'extrémité"	Permet la spécification d'un Point d'extrémité sans passer par le *Jeu de services "Découverte"*.

Le *Jeu de services "Session"* comprend plusieurs *ConformanceUnits* (voir Tableau 4). Les services CreateSession, ActivateSession et CloseSession sont pris en charge comme une unité individuelle. Tous les *Serveurs* et *Clients* fournissent cette fonctionnalité.

Tableau 4 – Services Session

Catégorie	Intitulé	Description
Serveur	*Session* "Comportement *Service* général"	Met en œuvre le comportement de *Service* de base. Ceci inclut notamment: – la vérification du jeton d'authentification; – la transmission de la requestHandle dans les réponses; – la transmission des informations de diagnostic disponibles demandée, avec le paramètre 'returnDiagnostics'; – le respect du message timeoutHint.
Serveur	*Session* "Base"	Prend en charge le *Jeu de services "Session"* (CreateSession, ActivateSession, CloseSession) à l'exception de l'utilisation d'ActivateSession pour changer l'utilisateur de la *Session*. Ceci inclut le traitement correct de tous les paramètres fournis. Pour les services CreateSession et ActivateSession, si le SecurityMode = None, alors: 1) le *Certificat* d'application et le Nonce sont facultatifs; 2) les signatures sont nulles/vides. Ces détails sont décrits dans l'IEC 62541-4.
Serveur	*Session* "Modification d'utilisateur"	Prend en charge l'utilisation d'ActivateSession pour modifier l'utilisateur de la *Session*.
Serveur	*Session* "Annuler"	Prend en charge le *Service* Annuler pour annuler les requêtes en suspens.
Serveur	*Session* "Minimum 1"	Prend en charge au moins 1 *session* (total).
Serveur	*Session* "Minimum 2 en parallèle"	Prend en charge au moins 2 Sessions en parallèle (total pour tous les *Clients*).
Serveur	*Session* "Minimum 50 en parallèle"	Prend en charge au minimum 50 Sessions en parallèle (total pour tous les *Clients*).
Serveur	Session "Invocation Sans Session"	Définit la prise en charge du Service "SessionlessInvoke" défini dans l'IEC 62541-4 pour traiter l'un des Services (comme Lecture/Ecriture, Parcourir ou Appel) qui sont conçus pour l'Invocation Sans Session.
Client	*Session Client* "Comportement de *Service* général"	Met en œuvre le comportement de *Service* de base. Ceci inclut notamment: – l'intégration du jeton d'authentification correct de la *Session*; – la création d'une requestHandle si nécessaire; – la demande d'informations de diagnostic avec le paramètre 'returnDiagnostics'; – l'évaluation du serviceResult et des résultats d'exploitation.
Client	*Session Client* "Base"	Utilise le *Jeu de services "Session"* (CreateSession, ActivateSession, CloseSession) à l'exception de l'utilisation d'ActivateSession pour changer l'utilisateur de la *Session*. Ceci inclut le traitement correct de tous les paramètres fournis. Pour les services CreateSession et ActivateSession, si le SecurityMode = None, alors: 1) le *Certificat* d'application et le Nonce sont facultatifs; 2) les signatures sont nulles/vides.
Client	*Session Client* "Connexions multiples"	Prend en charge un nombre illimité de connexions (côté client) avec plusieurs *Serveurs*. S'il y a une limite pour le nombre de connexions, elle est du côté serveur. Peut comporter une limite de mémoire, mais aucune limite de capacité logicielle.
Client	*Session Client* "Renouveler les NodeIds"	Cette *ConformanceUnit* s'applique aux *Clients* qui autorisent les NodeIds conservés. Vérifie que le tableau d'espace de noms n'a pas été modifié pour les NodeIds que le *Client* a conservés et va réutiliser au-delà de la durée de vie de la *Session*. Si des modifications sont intervenues, le *Client* doit recalculer les Indices d'espace de noms des NodeIds respectifs.
Client	*Session Client* "Masquer"	Utilise ActivateSession pour changer l'utilisateur de la Session (masquage).
Client	*Session Client* "KeepAlive"	Formule des demandes périodiques de maintien de la *Session* active.

Catégorie	Intitulé	Description
Client	*Session Client* "Détecter un arrêt"	Lit ou surveille la Variable ServerStatus/State afin de reconnaître un arrêt potentiel du Serveur et de "nettoyer" les ressources.
Client	*Session Client* "Annuler"	Utilise le *Service* Annuler pour annuler les requêtes en suspens.
Client	*Session Client* "Reconnexion automatique"	Reconnexion automatique du *Client*, y compris: – ActivateSession avec nouveau SecureChannel si le SecureChannel existant n'est plus valide, mais que la *Session* est toujours valide; – la création d'une nouvelle *Session* uniquement si la *Session* n'est plus valide
Client	*Session Client* "Session unique"	Le *Client* doit interagir avec des *Serveurs* qui prennent en charge une seule *Session*.
Client	*Session Client* "Appels de *Services* SessionLess"	Définit l'utilisation du *Service* "SessionlessInvoke" défini dans l'IEC 62541-4 pour traiter l'un des *Services* (comme Lecture/Ecriture ou Parcourir) qui sont admis pour l'invocation sans session. L'IEC 62541-6 spécifie quels transports peuvent être utilisés et comment.

Le *Jeu de services Gestion* des *Nœuds* comprend plusieurs *ConformanceUnits* (voir Tableau 5). Les *Serveurs* peuvent fournir certains aspects de cette fonctionnalité; se reporter aux *Profils* catégorisés comme *Profils de serveur* pour plus d'informations. Les *Clients* peuvent prendre en charge certains aspects de cette fonctionnalité; se reporter aux *Profils* catégorisés comme *Profils de client* pour plus d'informations.

Tableau 5 – Services Gestion des Nœuds

Catégorie	Intitulé	Description
Serveur	Gestion des *Nœuds* "Ajouter des *Nœuds*"	Prend en charge le *Service* AddNodes pour ajouter un ou plusieurs *Nœuds* dans l'*AddressSpace* OPC UA.
Serveur	Gestion des *Nœuds* "Supprimer des *Nœuds*"	Prend en charge le *Service* DeleteNodes pour supprimer un ou plusieurs *Nœuds* de l'*AddressSpace* OPC UA.
Serveur	Gestion des *Nœuds* "Ajouter Référence"	Prend en charge le *Service* AddReferences pour ajouter une ou plusieurs *Références* à un ou plusieurs *Nœuds* dans l'*AddressSpace* OPC UA.
Serveur	Gestion des *Nœuds* "Supprimer des Références"	Prend en charge le *Service* DeleteReferences pour supprimer une ou plusieurs *Références* d'un *Nœud* dans l'*AddressSpace* OPC UA.
Client	Gestion des *Nœuds Client*	Utilise les *Services* Gestion des *Nœuds* pour ajouter ou supprimer des *Nœuds* et pour ajouter ou supprimer des *Références* dans l'*AddressSpace* OPC UA du *Serveur*.

Le *Jeu de services* Vue comprend plusieurs *ConformanceUnits* (voir Tableau 6). Tous les *Serveurs* prennent en charge certains aspects de ce groupe de conformité. Les *Clients* peuvent prendre en charge certains aspects de cette fonctionnalité; se reporter aux *Profils* catégorisés comme *Profils de client* pour plus d'informations.

Tableau 6 – Services Vue

Catégorie	Intitulé	Description
Serveur	Vue de base	Prend en charge le *Jeu de services* Vue (Browse, BrowseNext).
Serveur	Vue "TranslateBrowsePath"	Prend en charge le *Service* TranslateBrowsePathsToNodeIds
Serveur	Vue "RegisterNodes"	Prend en charge les *Services* RegisterNodes et UnregisterNodes comme méthode d'optimisation de l'accès aux *Nœuds* à utilisation répétitive dans l'*AddressSpace* OPC UA du *Serveur*.
Serveur	Vue "Point de continuation minimum 01"	Prend en charge au minimum 1 point de continuation par *Session*.
Serveur	Vue "Point de continuation minimum 05"	Prend en charge au minimum 5 points de continuation par *Session*. Ce nombre doit être pris en charge pour au moins la moitié des sessions minimales exigées.
Client	Vue *Client* "Parcourir" de base	Utilise les *Services* Parcourir et BrowseNext pour naviguer dans l'*AddressSpace* OPC UA du *Serveur*. Utilise le referenceTypeId et le nodeClassMask pour spécifier les *Références* nécessaires.
Client	Vue *Client* "Parcourir les *Nœuds* distants"	Le *Client* peut naviguer jusqu'aux nœuds ayant un NodeID étendu qui référence un *Serveur* différent du *Serveur* d'origine. Cela comprend la connexion automatique au *Serveur* distant. Il est acceptable que les informations de configuration du *Serveur* soient préconfigurées sur le *Client* et/ou que l'utilisateur soit invité à se connecter.
Client	Vue *Client* "Filtrage de base du ResultSet"	Utilise le paramètre ResultMask pour optimiser l'ensemble de résultats que le *Serveur* a à renvoyer.
Client	Vue *Client* "TranslateBrowsePath"	Utilise le *Service* TranslateBrowsePathsToNodeIds pour identifier les NodeIds pour lesquels un *Nœud* de départ et un BrowsePath sont connus. Effectue des opérations d'ensemble plutôt que des appels multiples lorsque cela est possible.
Client	Vue *Client* "Parcourir la Traduction des *Nœuds* distants"	Le *Client* peut traduire les chemins de navigation comprenant des nœuds avec un NodeID qui référence un *Serveur* différent du *Serveur* d'origine et les renvoyer dans le cadre du *Service* TranslateBrowsePathsToNodeIds. Il est acceptable que les informations de configuration du *Serveur* soient préconfigurées sur le *Client*.
Client	Vue *Client* "RegisterNodes"	Utilise le *Service* RegisterNodes pour optimiser l'accès aux *Nœuds* à utilisation répétitive. Utilise le Service UnregisterNodes lorsque les *Nœuds* ne sont plus utilisés.

Le *Jeu de services Attribut* comprend plusieurs *ConformanceUnits* (voir Tableau 7). La majorité du *Jeu de services Attribut* constitue une fonctionnalité principale de l'OPC UA et est de ce fait pris en charge par la plupart des *Serveurs*. La plupart des *Clients* prennent également en charge certains aspects du *Jeu de services Attribut*.

Tableau 7 – Services Attribut

Catégorie	Intitulé	Description
Serveur	*Attribut* Lecture	Prend en charge le *Service* Lecture afin de déchiffrer un ou plusieurs *Attributs* d'un ou plusieurs *Nœuds*. Ceci inclut la prise en charge du paramètre IndexRange afin de déchiffrer un seul élément ou un ensemble d'éléments lorsque la valeur d'*Attribut* constitue une matrice.
Serveur	*Attribut* Lecture Complexe	Prend en charge la lecture et le codage de Valeurs avec des DataTypes structurés.
Serveur	*Attribut* Ecriture des Valeurs	Prend en charge l'écriture de valeurs à un ou plusieurs *Attributs* d'un ou plusieurs *Nœuds*.
Serveur	*Attribut* Ecriture Complexe	Prend en charge l'écriture et le codage de Valeurs avec DataTypes structurés.
Serveur	*Attribut* Ecriture de StatusCode & Horodatage	Prend en charge l'écriture du StatusCode et des Horodatages associés à la Valeur.
Serveur	*Attribut* Ecriture d'Indices	Prend en charge le paramètre IndexRange pour écrire un seul élément ou un ensemble d'éléments lorsque la valeur d'*Attribut* constitue une matrice et que les mises à jour partielles sont admises pour cette matrice.
Serveur	*Attribut* Codage alternatif	Prend en charge le codage alternatif des données lors de la lecture des *Attributs* de la Valeur. Par défaut, chaque *Serveur* doit prendre en charge le codage des données du *Profil* de Pile réellement utilisé (c'est-à-dire binaire avec le Codage UA binaire et XML avec le Codage XML). Cette *ConformanceUnit* (lorsqu'elle est prise en charge) spécifie que l'autre codage des données est également pris en charge.
Serveur	*Attribut* Lecture d'Historique	Prend en charge le *Service* HistoryRead. Les détails des aspects utilisés par ce service sont décrits par des *ConformanceUnits* supplémentaires, mais il importe qu'au moins l'une des unités suivantes soit prise en charge: ReadRaw, ReadProcessed, ReadModified, ReadAtTime ou ReadEvents.
Serveur	*Attribut* Mise à jour de l'Historique	Prend en charge le Service HistoryUpdate. Les détails des caractéristiques prises en charge de ce service sont décrits par des *ConformanceUnits* supplémentaires, mais il importe qu'au moins l'une des unités suivantes soit prise en charge: InsertData, InsertEvents, ReplaceData, ReplaceEvents, UpdateData, UpdateEvents, DeleteData, DeleteEvents ou DeleteAtTime.
Client	*Attribut Client* Lecture de base	Utilise le *Service* Lecture pour lire un ou plusieurs *Attributs* de un ou plusieurs *Nœuds*. Ceci inclut l'utilisation d'un IndexRange pour choisir un seul élément ou un ensemble d'éléments lorsque la valeur d'*Attribut* constitue une matrice. Les *Clients* doivent effectuer des opérations d'ensemble lorsque cela est possible, afin de réduire le nombre d'invocations de *Service*.
Client	*Attribut Client* Accès à l'*Attribut* des *Nœuds* distants	Le *Client* peut extraire les attributs des nœuds ayant un Identificateur de NodeID qui référence un *Serveur* différent du *Serveur* d'origine. Cela exige une connexion au *Serveur* distant pour pouvoir y accéder (pas nécessairement affichée comme une connexion). Il est acceptable que les informations de configuration du *Serveur* soient préconfigurées sur le *Client*.
Client	*Attribut Client* Lecture avec Codage approprié	Cette *ConformanceUnit* fait référence à la capacité d'un *Client* à découvrir les codages disponibles et à en choisir un spécifique lorsqu'il appelle le *Service* Lecture.
Client	*Attribut Client* Lecture Complexe	Lecture et décodage de Valeurs avec des DataTypes structurés.
Client	*Attribut Client* Ecriture de base	Utilise le *Service* Ecriture pour écrire des valeurs à un ou plusieurs *Attributs* d'un ou plusieurs *Nœuds*. Ceci inclut l'utilisation d'un IndexRange pour choisir un seul élément ou un ensemble d'éléments lorsque la valeur d'*Attribut* constitue une matrice. Les *Clients* doivent effectuer des opérations d'ensemble lorsque cela est possible, afin de réduire le nombre d'invocations de *Service*.
Client	*Attribut Client* Ecriture Complexe	Ecriture et codage de Valeurs avec des DataTypes structurés.
Client	*Attribut Client* Ecriture Qualité & Horodatage	Utilise le *Service* Ecriture pour écrire également le StatusCode et/ou les Horodatages associés à une Valeur.

Catégorie	Intitulé	Description
Client	*Attribut Client* Lecture d'Historique	Le *Client* utilise le service HistoryRead. Les détails des aspects utilisés par ce service sont fournis par des *ConformanceUnits* supplémentaires, mais au moins une ou plusieurs des unités suivantes sont utilisées: ReadRaw, ReadAtTime, ReadProcessed, ReadModified ou ReadEvents.
Client	*Attribut Client* Mises à jour de l'Historique	Le *Client* utilise le service HistoryUpdate. Les détails de cette utilisation sont décrits par des *ConformanceUnits* supplémentaires, mais il importe qu'au moins l'une des unités suivantes soit fournie: InsertData, InsertEvents, ReplaceData, ReplaceEvents, UpdateData, UpdateEvents, DeleteData oo DeleteEvents ou DeleteAtTime.

Le *Jeu de services Méthode* est constitué de *ConformanceUnits* (voir Tableau 8). Les *ConformanceUnits* principales assurent la prise en charge de la fonctionnalité d'appel. Les *Serveurs* peuvent fournir certains aspects de cette fonctionnalité; se reporter aux *Profils* catégorisés comme *Profils de serveur* pour plus d'informations. Les *Clients* peuvent prendre en charge certains aspects de cette fonctionnalité; se reporter aux *Profils* catégorisés comme *Profils de client* pour plus d'informations.

Tableau 8 – Services Méthode

Catégorie	Intitulé	Description
Serveur	*Méthode* Appel	Prend en charge le *Service* Appel pour appeler (invoquer) une *Méthode* qui inclut la prise en charge des paramètres de *Méthode*.
Client	*Méthode Client* Appel	Utilise le *Service* Appel pour appeler une ou plusieurs Méthodes.

Le *Jeu de services MonitoredItems* comprend plusieurs *ConformanceUnits* (voir Tableau 9). Les *Serveurs* peuvent fournir certains aspects de cette fonctionnalité; se reporter aux *Profils* catégorisés comme *Profils de serveur* pour plus d'informations. Les *Clients* peuvent prendre en charge certains aspects de cette fonctionnalité; se reporter aux *Profils* catégorisés comme *Profils de client* pour plus d'informations.

Tableau 9 – Services Eléments surveillés

Catégorie	Intitulé	Description
Serveur	Contrôle de base	Prend en charge les *Services MonitoredItem* suivants: CreateMonitoredItems, ModifyMonitoredItems, DeleteMonitoredItems, SetMonitoringMode.
Serveur	Contrôle Modification de valeur	Prend en charge la création de *MonitoredItems* pour les modifications de valeurs d'*Attributs*. Ceci inclut la prise en charge d'IndexRange afin de choisir un seul élément ou un ensemble d'éléments lorsque la valeur *Attribut* constitue une matrice.
Serveur	Contrôle de Valeur Complexe	Prend en charge la surveillance et le codage de Valeurs avec des DataTypes structurés.
Serveur	Filtre de Bande Morte d'Eléments Surveillés	Prend en charge un Filtre de bande morte absolu comme DataChangeFilter pour les types de données numériques.
Serveur	Contrôle de Filtre d'Agrégat	Prend en charge les filtres d'Agrégat pour les *MonitoredItems*. Le résultat de ces *ConformanceUnits* inclut une liste d'Agrégats qui sont pris en charge en tant que partie du *Certificat de Profil*.
Serveur	Contrôle de Codage alternatif	Prend en charge le codage alternatif lors de la surveillance des *Attributs* de la valeur. Par défaut, chaque *Serveur* doit prendre en charge le codage du *Profil* de Pile réellement utilisé (c'est-à-dire binaire avec le Codage UA binaire et XML avec le Codage XML). Cette *ConformanceUnit* (lorsqu'elle est prise en charge) spécifie que l'autre codage des données est également pris en charge.
Serveur	Contrôle de 2 éléments	Prend en charge au moins 2 *MonitoredItems* par *Abonnement*, où la taille de chaque *MonitoredItem* est au moins égale à la taille du Double.
Serveur	Contrôle de 10 éléments	Prend en charge au moins 10 *MonitoredItems* par *Abonnement*, où la taille de chaque *MonitoredItem* est au moins égale à la taille du Double.
Serveur	Contrôle de 100 éléments	Prend en charge au moins 100 *MonitoredItems* par *Abonnement*. Ce nombre doit être pris en charge pour au moins la moitié des Abonnements exigés, pour la moitié des Sessions exigées.
Serveur	Contrôle de 500 éléments	Prend en charge au moins 500 *MonitoredItems* par *Abonnement*. Ce nombre doit être pris en charge pour au moins la moitié des Abonnements exigés, pour la moitié des Sessions exigées.
Serveur	Contrôle de QueueSize_1	Cette *ConformanceUnit* n'exige pas la mise en file d'attente lorsque plusieurs modifications de valeurs surviennent pendant une période d'édition. Autrement dit, la dernière modification est envoyée dans la *Notification*.
Serveur	Contrôle de MinQueueSize_02	Prend en charge au moins 2 entrées de file d'attente pour les *MonitoredItems*. Les *Serveurs* adaptent souvent la taille de la file d'attente au nombre réel de *MonitoredItems*. Toutefois, il est prévu que les *Serveurs* prennent en charge cette taille de file d'attente minimale pour au moins un tiers des *MonitoredItems* pris en charge.
Serveur	Contrôle de MinQueueSize_05	Prend en charge au moins 5 entrées de file d'attente pour les *MonitoredItems*. Les *Serveurs* adaptent souvent la taille de la file d'attente au nombre réel de *MonitoredItems*. Toutefois, il est prévu que les *Serveurs* prennent en charge cette taille de file d'attente minimale pour au moins un tiers des *MonitoredItems* pris en charge.
Serveur	Contrôle de QueueSize_ServerMax	Cette *ConformanceUnit* s'applique aux Evénements. Lorsque le *Client* effectue une demande de queuesize=MAXUInt32, le *Serveur* doit retourner la taille maximale de la file d'attente qu'il peut prendre en charge pour les notifications d'événement en tant que revisedQueueSize.
Serveur	Contrôle de Déclenchement	Prend en charge le *Service* SetTriggering pour créer et/ou supprimer les liens actifs d'un élément déclencheur.

Catégorie	Intitulé	Description
Serveur	Contrôle d'Evénements	Prend en charge la création des *MonitoredItems* pour un "*Attribut EventNotifier*" à des fins de *Notification d'Evénements*. L'abonnement inclut la prise en charge d'un filtre qui comporte des opérandes SimpleAttribute et une liste sélective d'Opérateurs. La liste des Opérateurs inclut: Equals, IsNull, GreaterThan, LessThan, GreaterThanorEqual, LessThanorEqual, Like, Not, Between, InList, And, Or, Cast, BitwiseAnd, BitwiseOr.
Serveur	Contrôle de Filtre d'*Evénements* complexes	Prend en charge l'opérateur de filtre "TypeOf" pour les *Evénements* complexes.
Client	Contrôle *Client* Modification de valeur	Utilise le *Jeu de services MonitoredItem* pour enregistrer les éléments propres aux modifications de la valeur d'*Attribut*. Utiliser CreateMonitoredItems pour enregistrer la séquence *Nœud/Attribut*. Définir l'intervalle d'échantillonnage, le Filtre de Bande Morte et le mode de mise en file d'attente appropriés. Utilise le mode désactivation/activation de préférence au mode suppression et recréation d'un *MonitoredItem*. Exécuter des opérations d'ensemble de préférence plutôt que des requêtes de service individuel pour réduire les volumes de communication.
Client	Contrôle *Client* de Valeur Complexe	Surveille et décode les Valeurs avec des DataTypes structurés.
Client	Contrôle *Client* de Filtre de Bande Morte	Utilise les filtres de Bande Morte Absolus pour les abonnements.
Client	Contrôle *Client* par Indice	Utiliser IndexRange pour choisir un seul élément ou un ensemble d'éléments lorsque la valeur d'*Attribut* constitue une matrice.
Client	Contrôle *Client* de Filtre d'Agrégat	Utilise des filtres d'Agrégat pour les Abonnements.
Client	Contrôle *Client* d'Evénements	Utilise le *Jeu de services MonitoredItem* pour créer les *MonitoredItems* destinés aux notifications d'*événements*.
Client	Contrôle *Client* de Filtre d'Evénements	Utilise le filtre d'*Evénement* pour appeler le *Service* CreateMonitoredItems afin de filtrer les Evénements souhaités et de choisir les colonnes à prévoir pour chaque *Notification* d'*Evénement*.
Client	Contrôle *Client* de Filtre d'Evénements complexes	Utilise l'opérateur de filtre "TypeOf'" pour les *Evénements* complexes.
Client	Contrôle *Client* de Modification	Utilise le *Service* ModifyMonitoredItems pour modifier les paramètres de configuration. Utilise le *Service* SetMonitoringMode pour désactiver/activer l'échantillonnage et/ou l'édition.
Client	Contrôle *Client* de Déclenchement	Utilise le Modèle de déclenchement si certains éléments doivent être consignés uniquement en cas de déclenchement d'autres éléments spécifiques. Utilise le mode de surveillance approprié pour ces éléments. Utilise le *Service* SetTriggering pour relier ces éléments à l'élément déclencheur.

Le *Jeu de services Subscription* comprend plusieurs *ConformanceUnits* (voir Tableau 10). Les *Serveurs* peuvent fournir certains aspects de cette fonctionnalité; se reporter aux *Profils* catégorisés comme *Profils de serveur* pour plus d'informations. Les *Clients* peuvent prendre en charge certains aspects de cette fonctionnalité; se reporter aux *Profils* catégorisés comme *Profils de client* pour plus d'informations.

Tableau 10 – Services Abonnement

Catégorie	Intitulé	Description
Serveur	*Abonnement* de base	Prend en charge les *Services Abonnement* suivants: CreateSubscription, ModifySubscription, DeleteSubscriptions, Editer, Rééditer et SetPublishingMode
Serveur	*Abonnement* Minimum 1	Prend en charge au moins 1 *Abonnement* par *Session*. Ce nombre doit être pris en charge pour l'ensemble des sessions minimales exigées.
Serveur	*Abonnement* Minimum 02	Prend en charge au moins 2 *Abonnements* par *Session*. Ce nombre doit être pris en charge pour au moins la moitié des sessions minimales exigées.
Serveur	*Abonnement* Minimum 05	Prend en charge au moins 5 *Abonnements* par *Session*. Ce nombre doit être pris en charge pour au moins la moitié des sessions minimales exigées.
Serveur	*Abonnement* Editer Min 02	Prend en charge au moins 2 demandes de *Service* "Editer" par *Session*. Ce nombre doit être pris en charge pour l'ensemble des sessions minimales exigées. La prise en charge d'une file d'attente pour la retransmission du NotificationMessage n'est pas exigée; si elle n'est pas disponible, le *Service* "Rééditer" renvoie Bad_MessageNotAvailable.
Serveur	*Abonnement* Editer Min 05	Prend en charge au moins 5 demandes de *Service* "Editer" par *Session*. Ce nombre doit être pris en charge pour au moins la moitié des sessions minimales exigées. Prend en charge au minimum le nombre de requêtes d'édition par session comme taille de la file de retransmission du NotificationMessage à des fins de réédition.
Serveur	*Abonnement* Editer Min 10	Prend en charge au moins 10 demandes de *Service* "Editer" par *Session*. Ce nombre doit être pris en charge pour au moins la moitié des sessions minimales exigées. Prend en charge au minimum le nombre de requêtes d'édition par session comme taille de la file de retransmission du NotificationMessage à des fins de réédition.
Serveur	*Abonnement* Politique d'élimination d'édition	Respecte la politique spécifiée d'élimination des demandes de *Service* "Editer". Si le nombre maximal des demandes de *Service* "Editer" est dans une file d'attente et si une nouvelle demande de *Service* "Editer" se présente, la demande la "plus ancienne" doit être éliminée en renvoyant l'erreur appropriée.
Serveur	*Abonnement* Transfert	Prend en charge le *Service* TransferSubscriptions pour un transfert d'*Abonnement* d'une *Session* à l'autre.
Serveur	*Abonnement* Durable	Prend en charge la configuration des Abonnements en mode durable. Ce mode exige que les données et les événements collectés soient stockés et distribués même si un *Client* est déconnecté depuis longtemps ou si le *Serveur* a été redémarré. Prend en charge l'une des *ConformanceUnits* "*Abonnement* Durable StorageLevel nnn".
Serveur	*Abonnement* Durable StorageLevel Faible	Prend en charge au moins 20 éléments surveillés avec une file d'attente de 10 000 pour chaque élément et lorsque la taille de chaque *MonitoredItem* est au moins égale à la taille du Double. Une capacité de stockage 200 000 valeurs est exigée pour le DataType Double.
Serveur	*Abonnement* Durable StorageLevel Moyen	Prend en charge au moins 100 éléments surveillés avec une file d'attente de 50 000 pour chaque élément et lorsque la taille de chaque *MonitoredItem* est au moins égale à la taille du Double. Une capacité de stockage de 5 millions de valeurs est exigée pour le DataType Double.
Serveur	*Abonnement* Durable StorageLevel Elevé	Prend en charge au moins 2 000 éléments surveillés avec une file d'attente de 200 000 pour chaque élément et lorsque la taille de chaque *MonitoredItem* est au moins égale à la taille du Double. Une capacité de stockage de 400 millions de valeurs est exigée pour le DataType Double.

Catégorie	Intitulé	Description
Client	*Abonnement Client* de base	Utilise le *Jeu de services Subscription* et *MonitoredItems* comme moyen efficace de détecter les modifications de valeurs d'*Attributs* et/ou de recevoir les occurrences d'*Evénements*. Définir les intervalles d'édition appropriés et maintenir actives les notifications et la durée de vie totale d'*Abonnement*. Fournir un nombre suffisant de demandes "Editer" au *Serveur* de manière à pouvoir transmettre les *Notifications* à chaque expiration d'un temporisateur d'édition. Acquitter des *Notifications* reçues avec les requêtes d'édition ultérieures.
Client	*Abonnement Client* Repli	Le *Client* doit interagir avec les *Serveurs* qui ne prennent pas en charge les Abonnements, ou qui ont atteint leur nombre limite d'*Abonnements*, pour la Surveillance au moyen du *Service* Lecture.
Client	*Abonnement* Réédition *Client*	Evalue le nombre de séquences des *Notifications* afin de détecter les *Notifications* perdues. Utilise "Rééditer" pour demander les *Notifications* manquantes.
Client	*Abonnement* Modification *Client*	Permet la modification de la configuration d'*Abonnement* en utilisant le *Service* ModifySubscription.
Client	*Abonnement Client* TransferSubscriptions	Le *Client* prend en charge les *Abonnements* de transfert d'autres *Clients*. Cette *ConformanceUnit* est utilisée dans le cadre des *Clients* redondants.
Client	*Abonnement Client* Multiple	Utilise les différents abonnements pour réduire la charge utile des *Notifications* individuelles.
Client	*Abonnement* Edition configurable *Client*	Transmet plusieurs demandes de *Service* "Editer" afin de s'assurer que le *Serveur* est toujours capable de transmettre des *Notifications*. Le nombre de demandes parallèles de *Service* "Editer" par *Session* doit être configurable.
Client	*Abonnement* Durable *Client*	Utilise les abonnements durables.

5.3 Caractéristiques relatives au transport et à la communication

Le Tableau 11 décrit les *ConformanceUnits* relatives à la sécurité. Toutes ces *ConformanceUnits* s'appliquent de manière égale aux *Clients* et aux *Serveurs*, lorsqu'un *Client* utilise l'unité relative à la sécurité et un *Serveur* prend en charge l'utilisation de cette dernière. Ces éléments sont définis en détail dans l'IEC 62541-6. Il est recommandé que le *Serveur* et le *Client* prennent en charge le plus grand nombre possible de ces options afin d'obtenir des niveaux d'interopérabilité plus élevés. Il revient à l'administrateur de déterminer quelle *ConformanceUnit* est présentée dans une application *Serveur* ou *Client* déployée donnée.

Tableau 11 – Sécurité

Catégorie	Intitulé	Description
Sécurité	Sécurité Nom d'Utilisateur Mot de Passe	Le Serveur prend en charge les combinaisons Nom d'utilisateur/Mot de passe. Le jeton est chiffré si la politique de sécurité de la Politique relative aux jetons d'utilisateurs ou la politique de sécurité du point d'extrémité l'exige. Un jeton non chiffré exige soit le chiffrement du message, soit un dispositif hors de l'OPC UA permettant de sécuriser le jeton d'identité afin qu'il ne puisse pas être récupéré par l'écoute des communications (sniffing). Un transport sécurisé comme un VPN serait une option.
Sécurité	Sécurité utilisateur X509	Le *Serveur* prend en charge une paire de clés publique/privée pour l'identité de l'utilisateur. Il importe que l'utilisation de cette caractéristique puisse être activée ou désactivée par un administrateur.
Sécurité	Sécurité utilisateur IssuedToken Kerberos	Le *Serveur* prend en charge un jeton de *Serveur* Kerberos pour l'Identité de l'Utilisateur. Il importe que l'utilisation de cette caractéristique puisse être activée ou désactivée par un administrateur. L'utilisation de ce jeton est définie dans la Documentation de jeton Kerberos. Le jeton est chiffré si la politique de sécurité de la Politique relative aux jetons d'utilisateurs ou la politique de sécurité du point d'extrémité l'exige. Un jeton non chiffré exige soit le chiffrement du message, soit un dispositif hors de l'OPC UA permettant de sécuriser le jeton d'identité afin qu'il ne puisse pas être récupéré par l'écoute des communications (sniffing). Un transport sécurisé comme un VPN serait une option.
Sécurité	Sécurité utilisateur IssuedToken Kerberos Windows	Le *Serveur* prend en charge la mise en œuvre Windows des jetons Kerberos. Cette *ConformanceUnit* ne s'applique que si la "Sécurité utilisateur IssuedToken Kerberos" est prise en charge. Le jeton est chiffré si la politique de sécurité de la Politique relative aux jetons d'utilisateurs ou la politique de sécurité du point d'extrémité l'exige. Un jeton non chiffré exige soit le chiffrement du message, soit un dispositif hors de l'OPC UA permettant de sécuriser le jeton d'identité afin qu'il ne puisse pas être récupéré par l'écoute des communications (sniffing). Un transport sécurisé comme un VPN serait une option.
Sécurité	Sécurité utilisateur IssuedToken JWT	Le *Serveur* prend en charge un jeton JWT (JSON Web Token) pour l'Identité de l'Utilisateur. L'IEC 62541-6 décrit OAuth2 et les jetons JWT plus en détail. Il importe que l'utilisation de cette caractéristique puisse être activée ou désactivée par un administrateur. Le jeton est chiffré si la politique de sécurité de la Politique relative aux jetons d'utilisateurs ou la politique de sécurité du point d'extrémité l'exige. Un jeton non chiffré exige soit le chiffrement du message, soit un dispositif hors de l'OPC UA permettant de sécuriser le jeton d'identité afin qu'il ne puisse pas être récupéré par l'écoute des communications (sniffing). Un transport sécurisé comme un VPN serait une option.
Sécurité	Sécurité utilisateur Anonyme	Le *Serveur* prend en charge l'accès Anonyme. Il importe que l'utilisation de cette caractéristique puisse être activée ou désactivée par un administrateur. L'accès Anonyme doit être désactivé par défaut.
Sécurité	Sécurité utilisateur Nom d'utilisateur/Mot de passe *Client*	Un *Client* utilise une combinaison Nom d'utilisateur/Mot de passe. Le jeton est chiffré si la politique de sécurité de la Politique relative aux jetons d'utilisateurs ou la politique de sécurité du point d'extrémité l'exige. Un jeton non chiffré exige soit le chiffrement du message, soit un dispositif hors de l'OPC UA permettant de sécuriser le jeton d'identité afin qu'il ne puisse pas être récupéré par l'écoute des communications (sniffing). Un transport sécurisé comme un VPN serait une option.
Sécurité	Sécurité utilisateur X509 *Client*	Un *Client* utilise une paire de clés publique/privée pour l'identité de l'utilisateur. Cela comprend tous les problèmes de validation et de confiance associés à un certificat.
Sécurité	Sécurité utilisateur IssuedToken Kerberos *Client*	Un *Client* utilise un jeton de *Serveur* Kerberos. L'utilisation de ce jeton est définie dans la documentation Kerberos. Le jeton est chiffré si la politique de sécurité de la Politique relative aux jetons d'utilisateurs ou la politique de sécurité du point d'extrémité l'exige. Un jeton non chiffré exige soit le chiffrement du message, soit un dispositif hors de l'OPC UA permettant de sécuriser le jeton d'identité afin qu'il ne puisse pas être récupéré par l'écoute des communications (sniffing). Un transport sécurisé comme un VPN serait une option.

Catégorie	Intitulé	Description
Sécurité	Sécurité utilisateur IssuedToken Kerberos Windows Client	Un *Client* utilise la mise en œuvre Windows des jetons Kerberos. Cette *ConformanceUnit* ne s'applique que si la "Sécurité utilisateur *Client* IssuedToken Kerberos" est prise en charge. Le jeton est chiffré si la politique de sécurité de la Politique relative aux jetons d'utilisateurs ou la politique de sécurité du point d'extrémité l'exige. Un jeton non chiffré exige soit le chiffrement du message, soit un dispositif hors de l'OPC UA permettant de sécuriser le jeton d'identité afin qu'il ne puisse pas être récupéré par l'écoute des communications (sniffing). Un transport sécurisé comme un VPN serait une option.
Sécurité	Sécurité utilisateur IssuedToken JWT *Client*	Un *Client* utilise un jeton JWT (JSON Web Token) pour l'identité de l'utilisateur. L'IEC 62541-6 décrit OAuth2 et les jetons JWT plus en détail. Le jeton est chiffré si la politique de sécurité de la Politique relative aux jetons d'utilisateurs ou la politique de sécurité du point d'extrémité l'exige. Un jeton non chiffré exige soit le chiffrement du message, soit un dispositif hors de l'OPC UA permettant de sécuriser le jeton d'identité afin qu'il ne puisse pas être récupéré par l'écoute des communications (sniffing). Un transport sécurisé comme un VPN serait une option.
Sécurité	Sécurité Jeton d'utilisateur non valide	Les *Serveurs* doivent prendre des mesures appropriées pour se protéger des attaques contre les jetons d'identité d'utilisateur. Par hypothèse, ces attaques correspondent à des tentatives de connexion répétées au moyen de jetons d'identité d'utilisateur non valides. Voir *Service* ActivateSession dans l'IEC 62541-4.
Sécurité	Sécurité utilisateur Politique relative aux jetons JWT	Le Serveur prend en compte un ou plusieurs Points d'extrémité avec une UserTokenPolicy comprenant une IssuerEndpointUrl pour les jetons JWT définis dans l'IEC 62541-6. Pour les jetons JWT, l'issuerEndpointUrl est un objet JSON incluant tous les paramètres qui définissent l'AuthorizationService. Selon la Politique relative aux jetons JWT, le Serveur doit prendre en charge au moins l'une des Unités de Conformité des Profils d'Autorité suivantes. Les URI définies dans la ConformanceUnit doivent être présentées dans le champ authorityProfileURI de la Politique relative aux jetons JWT.
Sécurité	Sécurité utilisateur Politique relative aux jetons JWT *Client*	Le Client comprend et utilise la définition du Service d'Autorisation incluse dans l'UserTokenPolicy pour les jetons JWT renvoyée avec GetEndpoints. Elle doit prendre en charge au moins l'une des Unités de Conformité de Profil d'Autorité suivantes. Les URI définies dans la ConformanceUnit figurent dans le champ authorityProfileURI de la Politique relative aux jetons JWT présentée dans les Points d'extrémité des Serveurs.
Sécurité	*Profil* d'autorité OAuth2	Cette unité indique la prise en charge de OAuth2 plutôt que HTTPS pour demander les jetons d'accès. L'URI pour les interactions avec cette autorité est "http://opcfoundation.org/UA/Authorization#OAuth2"
Sécurité	*Profil* d'autorité OPC UA	Cette unité indique la prise en charge des *Méthodes* OPC UA définies dans l'IEC 62541-12 pour demander les jetons d'accès. L'URI pour les interactions avec cette autorité est "http://opcfoundation.org/UA/Authorization#OPCUA"
Sécurité	*Profil* d'Autorité Fournisseur d'Identité Azure	Cette unité indique la prise en charge du Fournisseur d'Identité Azure pour demander les jetons d'accès. L'URI pour les interactions avec cette autorité est "http://opcfoundation.org/UA/Authorization#Azure"
Sécurité	Sécurité Validation de *Certificat*	Un Certificat est validé comme spécifié dans l'IEC 62541-4. Ceci inclut, entre autres, l'examen de la structure et de la signature. Permet la suppression de certaines erreurs de validation par directive administrative.
Sécurité	Sécurité Certificat ApplicationInstance par défaut	Une application, une fois installée, a par défaut un ApplicationInstanceCertificate valide. L'ApplicationInstanceCertificate par défaut doit soit être créé dans le cadre de l'installation, soit les instructions d'installation doivent décrire de façon explicite le processus pour créer et appliquer un ApplicationInstanceCertificate par défaut à l'application.

Catégorie	Intitulé	Description
Sécurité	Sécurité – Pas d'Authentification d'Application	Le *Serveur* prend en charge la capacité de n'être configuré pour aucune authentification d'application, seulement l'authentification de l'utilisateur et le mode de cryptage/signature normal: – le *Serveur* est configuré pour accepter tous les certificats; – les *Certificats* sont utilisés uniquement pour la sécurité des messages (signature et cryptage); – le niveau Utilisateurs est utilisé pour l'authentification.
Sécurité	Sécurité Politique exigée	Prend en charge au moins la Politique de sécurité [A] et la Politique de sécurité [B]. La prise en charge de plusieurs Politiques de sécurité (même obsolètes) est recommandée. Cela garantit une meilleure interopérabilité tout en permettant à l'utilisateur de choisir le niveau de sécurité exigé. Les Politiques de sécurité obsolètes ne doivent pas être activées/utilisables sans intervention de l'administrateur.
Sécurité	Security None CreateSession ActivateSession	Lorsque SecurityPolicy = None, les Services CreateSession et ActivateSession permettent une signature NULLE/vide et n'exigent ni *Certificat* d'application, ni Nonce.
Sécurité	Security None CreateSession ActivateSession 1.0	Le *Client* peut se connecter aux *Serveurs* qui exigent de transmettre un certificat lors de l'établissement de la Session. Dans ce cas, le *Client* effectue une première tentative sans *Certificat* et, en cas d'échec, présente un *Certificat*.
Sécurité	Sécurité TLS Générale	Cette *ConformanceUnit* indique qu'au moins un des *Profils* de sécurité transport pour TLS (sécurité de niveau de transport) est pris en charge par cette application. Elle est utilisée dans les *Profils* de transport TLS, mais le choix du profil de sécurité transport est facultatif. Le profil de sécurité réellement utilisé est par défaut le plus sécurisé.
Sécurité	Sécurité TLS_RSA avec AES_256_CBC_SHA256	La connexion est établie à l'aide de TLS_RSA_WITH_AES_256_CBC_SHA256. Ses caractéristiques sont MinAsymmetricKeyLength – 2048, MaxAsymmetricKeyLength – 4096, AsymmetricSignatureAlgorithm – RSA_SHA256. (TLS 1.2)
Sécurité	Sécurité TLS_DHE_RSA avec AES_nnn_CBC_SHA256	La connexion est établie à l'aide de TLS_DHE_RSA avec AES_128_CBC_SHA256 ou TLS_DHE_RSA avec AES_256_CBC_SHA256. Ses caractéristiques sont MinAsymmetricKeyLength – 2048, MaxAsymmetricKeyLength – 4096, CertificateSignatureAlgorithm – RSA_SHA256. (TLS 1.2). Les *Clients* et les *Serveurs* doivent prendre en charge les deux algorithmes.
Sécurité	Sécurité Cryptage exigé	Un cryptage utilisant les algorithmes fournis dans la suite d'algorithmes de sécurité est exigé.
Sécurité	Sécurité Signature exigée	Une signature utilisant les algorithmes fournis dans la suite d'algorithmes de sécurité est exigée.
Sécurité	Synchronisation Temporelle Sécurité – Configuration	L'application prend en charge la configuration d'un décalage d'horloge acceptable.
Sécurité	Synchronisation temporelle Sécurité – Prise en charge NTP/OS	L'application prend en charge la synchronisation temporelle, soit par la mise en œuvre du Protocole d'Heure Réseau (NTP – Network Time Protocol), soit par les caractéristiques d'un système d'exploitation normalisé.
Sécurité	Synchronisation temporelle Sécurité – Prise en charge UA	Une application utilise les réponses de l'horodatage d'en-tête fournies par une source connue configurée, telle qu'un *Serveur "Découverte"*, afin de synchroniser le temps sur l'application et afin que cette synchronisation temporelle se produise régulièrement. L'utilisation de cette TimeSyncing peut être configurée.

Catégorie	Intitulé	Description
Sécurité	Sécurité Administration	Permet la configuration des éléments suivants relatifs à la sécurité (si applicables). * choisit la(les) politique(s) autorisée(s) d'identification de l'utilisateur (par exemple, Nom/Mot de passe de l'Utilisateur ou X509) * active/désactive la politique de sécurité "None" ou autres politiques de sécurité * active/désactive les points d'extrémité avec MessageSecurityMode SIGN ou SIGNANDENCRYPT. * définit les autorités de certification admises * définit la méthode de réaction aux Certificats inconnus * permet d'accepter tous les Certificats valides
Sécurité	Sécurité Administration – Schéma XML	Prend en charge le schéma XML défini par OPC UA pour l'importation et l'exportation des informations de configuration de sécurité. Ce schéma est spécifié dans l'IEC 62541-6.
Sécurité	Sécurité *Certificat* d'Administration	Autoriser l'administrateur du site à pouvoir attribuer un ApplicationInstanceCertificate spécifique au site, et s'il le souhaite, configurer une Autorité de Certification (CA) spécifique au site.
Sécurité	Sécurité Rôle de base *Serveur*	Prend en charge le Modèle d'information sur les Autorisations d'utilisateurs défini dans l'UA, Parties 3 et 5 (comme les rôles) et les *Attributs* RolePermissions et UserRolePermissions.
Sécurité	Sécurité Rôle bien connu	Prend en charge les rôles connus "ConfigureAdmin" et "SecurityAdmin" avec les permissions suggérées définies dans l'IEC 62541-3.
Sécurité	Sécurité Rôle *Serveur* IdentityManagement	Permet aux utilisateurs autorisés d'ajouter et/ou de supprimer des Identités dans les Rôles selon les *Méthodes* adéquates.
Sécurité	Sécurité Rôle *Serveur* Gestion	Permet aux utilisateurs autorisés de créer de nouveaux Rôles et/ou de supprimer des Rôles selon les *Méthodes* adéquates.
Sécurité	Sécurité Rôle *Serveur* Restreindre Applications	Prend en charge l'ajout d'applications à un Rôle selon les *Méthodes* adéquates, afin que seules ces applications puissent utiliser ce Rôle
Sécurité	Sécurité Rôle *Serveur* Restreindre les Points d'extrémité	Prend en charge l'ajout de Points d'extrémité à un Rôle selon les *Méthodes* adéquates. Avec cette restriction, un Rôle est appliqué seulement lorsqu'un *Client* se connecte au moyen de l'un de ces Points d'extrémité.
Sécurité	Sécurité Rôle *Serveur* DefaultRolePermissions	Permet aux utilisateurs autorisés de régler la *Propriété* DefaultRolePermissions pour certains NameSpaces. Les DefaultRolePermissions sont appliquées si aucune RolePermission n'est associée à un *Nœud*.
Sécurité	Sécurité Rôle *Serveur* RolePermissions	Permet aux utilisateurs autorisés de régler l'*Attribut* RolePermissions sur *Nœuds*.
Sécurité	Sécurité – Autorisation de Rôles *Serveur*	Restreindre l'accès selon les Rôles et permissions configurés.
Sécurité	Sécurité Rôle *Client* de base	Comprend et utilise le Modèle d'information sur l'Autorisation de l'utilisateur défini dans l'IEC 62541-5 et l'*Attribut* RolePermissions.
Sécurité	Sécurité Rôle *Client* Gestion	Prend en charge la création de nouveaux Rôles et l'ajout d'identités, ainsi que la suppression de Rôles ou d'Identités selon les *Méthodes* adéquates.
Sécurité	Sécurité Rôle *Client* Restreindre Applications	Utilise les *Méthodes* adéquates pour ajouter des applications à un Rôle, afin que seules ces applications puissent utiliser ce Rôle.
Sécurité	Sécurité Rôle *Client* Restreindre Points d'extrémité	Utilise les *Méthodes* adéquates pour ajouter des Points d'extrémité à un Rôle. Avec cette restriction, un Rôle est appliqué seulement lorsqu'un *Client* se connecte au moyen de l'un de ces Points d'extrémité.
Sécurité	Sécurité Rôle *Client* DefaultRolePermissions	Capacité de régler la *Propriété* DefaultRolePermissions pour certains NameSpaces. Les DefaultRolePermissions sont appliquées si aucune RolePermission n'est associée à un *Nœud*.
Sécurité	Sécurité Rôle *Client* RolePermissions	Prend en charge le réglage de l'*Attribut* RolePermissions sur des *Nœuds*.
Sécurité	Modèle Pull pour la Gestion Globale des *Certificats* et des TrustLists	Utilise les services de gestion de *Certificats* de l'IEC 62541-12 pour que le modèle Pull gère les *Certificats* d'instance d'application et les Listes de confiance, y compris des Listes de révocation.

Catégorie	Intitulé	Description
Sécurité	Modèle Push pour la Gestion Globale des *Certificats* et des TrustLists	Prend en charge les services de gestion de *Certificats* de l'IEC 62541-12 pour que le modèle Push gère les *Certificats* d'instance d'application et les Listes de confiance, y compris des Listes de révocation.
Sécurité	Modèle Pull pour le *Service* KeyCredential	Utilise les *Méthodes* sur une instance du KeyCredentialServiceType (modèle Pull) pour obtenir les KeyCredentials spécifiés dans l'IEC 62541-12.
Sécurité	Modèle Push pour le *Service* KeyCredential	Prend en charge le Modèle *Push* des *Services* KeyCredential de l'IEC 62541-12 pour obtenir les KeyCredentials. Cela comprend la prise en charge d'une ou plusieurs instances du KeyCredentialConfigurationType et des *Méthodes* pour mettre à jour ou supprimer les authentifiants.
Sécurité	Configuration du Service d'autorisation Serveur	Prend en charge les Types d'objets définis dans l'IEC 62541-12 afin de permettre la configuration des informations nécessaires pour accepter les Jetons d'accès lorsqu'ils sont présentés par le Client pendant l'établissement de la session. Les Jetons d'accès sont émis par les Services d'autorisation.
Sécurité	Service d'Autorisation Client	Utilise la méthode RequestAccessToken définie dans l'IEC 62541-12.
Sécurité	SymmetricSignatureAlgorithm_None	Cet algorithme ne s'applique pas.
Sécurité	SymmetricSignatureAlgorithm_HMAC-SHA1	Hachage avec clé défini dans https://tools.ietf.org/html/rfc2104. L'algorithme de hachage est SHA1 et il est décrit dans https://tools.ietf.org/html/rfc3174. L'URI est http://www.w3.org/2000/09/xmldsig#hmac-sha1 Il n'existe aucune exploitation connue utilisant SHA1 avec un hachage à clé. Toutefois, SHA1 a été cassé en 2017 et l'utilisation de cet algorithme n'est donc pas recommandée.
Sécurité	SymmetricSignatureAlgorithm_HMAC-SHA2-256	Hachage avec clé utilisé pour l'authentification des messages, défini dans https://tools.ietf.org/html/rfc2104. L'algorithme de hachage est SHA2 avec 256 bits et il est décrit dans https://tools.ietf.org/html/rfc4634.
Sécurité	SymmetricEncryptionAlgorithm_None	Cet algorithme ne s'applique pas.
Sécurité	SymmetricEncryptionAlgorithm_AES128-CBC	L'algorithme de chiffrement AES qui est défini dans http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf. Plusieurs blocs chiffrés au moyen du mode CBC décrit dans http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf. La taille de la clé est de 128 bits. La taille du bloc est de 16 octets. L'URI est http://www.w3.org/2001/04/xmlenc#aes128-cbc
Sécurité	SymmetricEncryptionAlgorithm_AES256-CBC	L'algorithme de chiffrement AES qui est défini dans http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf. Plusieurs blocs chiffrés au moyen du mode CBC décrit dans http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf. La taille de la clé est de 256 bits. La taille du bloc est de 16 octets. L'URI est http://www.w3.org/2001/04/xmlenc#aes256-cbc
Sécurité	SymmetricEncryptionAlgorithm_AES128-CTR	L'algorithme de chiffrement AES qui est défini dans http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf. Plusieurs blocs chiffrés au moyen du mode CTR décrit dans http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf. Le format du bloc compteur est défini dans https://tools.ietf.org/html/rfc3686. La taille de la clé est de 128 bits. La taille du bloc est de 16 octets. La longueur du nonce d'entrée est de 4 octets. L'URI est http://opcfoundation.org/UA/security/aes128-ctr

Catégorie	Intitulé	Description
Sécurité	SymmetricEncryptionAlgorithm_AES256-CTR	L'algorithme de chiffrement AES qui est défini dans http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197.pdf. La taille de la clé est de 256 bits. La taille du bloc est de 16 octets. Plusieurs blocs chiffrés au moyen du mode CTR décrit dans http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf. Le format du bloc compteur est défini dans https://tools.ietf.org/html/rfc3686. La taille de la clé est de 128 bits. La taille du bloc est de 16 octets. La longueur du nonce d'entrée est de 4 octets. L'URI est http://opcfoundation.org/UA/security/aes256-ctr.
Sécurité	AsymmetricSignatureAlgorithm_None	Cet algorithme ne s'applique pas.
Sécurité	AsymmetricSignatureAlgorithm_RSA- PKCS15-SHA1	L'algorithme de signature RSA qui est défini dans https://tools.ietf.org/html/rfc3447. Le schéma RSASSA-PKCS1-v1_5 est utilisé. L'algorithme de hachage est SHA1 et il est décrit dans https://tools.ietf.org/html/rfc3174. L'URI est http://www.w3.org/2000/09/xmldsig#rsa-sha1. SHA1 a été cassé en 2017 et il convient donc de ne pas utiliser cet algorithme.
Sécurité	AsymmetricSignatureAlgorithm_RSA-PKCS15-SHA2-256	L'algorithme de signature RSA qui est défini dans https://tools.ietf.org/html/rfc3447. Le schéma RSASSA-PKCS1-v1_5 est utilisé. L'algorithme de hachage est SHA2 avec 256 bits et il est décrit dans https://tools.ietf.org/html/rfc6234. L'URI est http://www.w3.org/2001/04/xmldsig-more#rsa-sha256.
Sécurité	AsymmetricSignatureAlgorithm_RSA-PSS -SHA2-256	L'algorithme de signature RSA qui est défini dans https://tools.ietf.org/html/rfc3447. Le schéma RSASSA-PSS est utilisé. L'algorithme de hachage est SHA2 avec 256 bits et il est décrit dans https://tools.ietf.org/html/rfc6234. L'algorithme de génération du masque utilise aussi SHA2 avec 256 bits. La longueur du grain de sel est 32 octets. L'URI est http://opcfoundation.org/UA/security/rsa-pss -sha2-256.
Sécurité	AsymmetricEncryptionAlgorithm_None	Cet algorithme ne s'applique pas.
Sécurité	AsymmetricEncryptionAlgorithm_RSA-PKCS15	L'algorithme de chiffrement RSA qui est défini dans https://tools.ietf.org/html/rfc3447. Le schéma RSAES-PKCS1-v1_5 est utilisé. L'URI est http://www.w3.org/2001/04/xmlenc#rsa-1_5. Le schéma RSAES-PKCS1-v1_5 a des faiblesses connues et il n'est pas recommandé.
Sécurité	AsymmetricEncryptionAlgorithm_RSA-OAEP-SHA1	L'algorithme de chiffrement RSA qui est défini dans https://tools.ietf.org/html/rfc3447. Le schéma RSAES-OAEP est utilisé. L'algorithme de hachage est SHA1 et il est décrit dans https://tools.ietf.org/html/rfc6234. L'algorithme de génération du masque utilise également SHA1. L'URI est http://www.w3.org/2001/04/xmlenc#rsa-oaep. Il n'existe aucune exploitation connue utilisant SHA1 avec RSAES-OAEP. Toutefois, SHA1 a été cassé en 2017 et l'utilisation de cet algorithme n'est donc pas recommandée.

Catégorie	Intitulé	Description
Sécurité	AsymmetricEncryptionAlgorithm_RSA-OAEP-SHA2-256	L'algorithme de chiffrement RSA qui est défini dans https://tools.ietf.org/html/rfc3447. Le schéma RSAES-OAEP est utilisé. L'algorithme de hachage est SHA2 avec 256 bits et il est décrit dans https://tools.ietf.org/html/rfc6234. L'algorithme de génération du masque utilise aussi SHA2 avec 256 bits. L'URI est http://opcfoundation.org/UA/security/rsa-oaep-sha2-256.
Sécurité	KeyDerivationAlgorithm_None	Cet algorithme ne s'applique pas.
Sécurité	KeyDerivationAlgorithm_P-SHA1	La fonction pseudoaléatoire P_SHA-1 est définie dans https://tools.ietf.org/html/rfc4346. L'URI est http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/dk/p_sha1. Il n'existe aucune exploitation connue utilisant SHA1 avec P-SHA-1. Toutefois, SHA1 a été cassé en 2017 et l'utilisation de cet algorithme n'est donc pas recommandée.
Sécurité	KeyDerivationAlgorithm_P-SHA2-256	La fonction pseudoaléatoire P_SHA256 est définie dans https://tools.ietf.org/html/rfc5246. L'URI est http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/dk/p_sha256.
Sécurité	CertificateSignatureAlgorithm_None	Cet algorithme ne s'applique pas.
Sécurité	CertificateSignatureAlgorithm_RSA-PKCS15-SHA2-256	L'algorithme de signature RSA qui est défini dans https://tools.ietf.org/html/rfc3447. Le schéma RSASSA-PKCS1-v1_5 est utilisé. L'algorithme de hachage est SHA2 avec 256 bits et il est décrit dans https://tools.ietf.org/html/rfc6234. L'algorithme SHA2 avec 384 ou 512 bits peut être utilisé à la place de SHA2 avec 256 bits. L'URI est http://www.w3.org/2001/04/xmldsig-more#rsa-sha256.
Sécurité	CertificateSignatureAlgorithm_RSA-PKCS15-SHA1	L'algorithme de signature RSA qui est défini dans https://tools.ietf.org/html/rfc3447. Le schéma RSASSA-PKCS1-v1_5 est utilisé. L'algorithme de hachage est SHA1 et il est décrit dans https://tools.ietf.org/html/rfc3174. L'URI est http://www.w3.org/2000/09/xmldsig#rsa-sha1. SHA1 a été cassé en 2017 et il convient donc de ne pas utiliser cet algorithme. L'algorithme SHA2 avec 244, 256, 384 ou 512 bits peut être utilisé à la place de SHA1. L'algorithme SHA2 est décrit dans https://tools.ietf.org/html/rfc6234.
Sécurité	SecurtyPolicy_None_Limits	DerivedSignatureKeyLength: 0
Sécurité	Aes128-Sha256-RsaOaep_Limits	-> DerivedSignatureKeyLength: 256 bits -> MinAsymmetricKeyLength: 2048 bits -> MaxAsymmetricKeyLength: 4096 bits -> SecureChannelNonceLength: 32 octets
Sécurité	Basic256Sha256_Limits	-> DerivedSignatureKeyLength: 256 bits -> MinAsymmetricKeyLength: 2048 bits -> MaxAsymmetricKeyLength: 4096 bits -> SecureChannelNonceLength: 32 octets

Catégorie	Intitulé	Description
Sécurité	Aes256-Sha256-RsaPss_Limits	-> DerivedSignatureKeyLength: 256 bits -> MinAsymmetricKeyLength: 2048 bits -> MaxAsymmetricKeyLength: 4096 bits -> SecureChannelNonceLength: 32 octets
Sécurité	Basic128Rsa15_Limits	-> DerivedSignatureKeyLength: 128 bits -> MinAsymmetricKeyLength: 1024 bits -> MaxAsymmetricKeyLength: 2048 bits -> SecureChannelNonceLength: 16 octets
Sécurité	Basic256_Limits	-> DerivedSignatureKeyLength: 192 bits -> MinAsymmetricKeyLength: 1024 bits -> MaxAsymmetricKeyLength: 2048 bits -> SecureChannelNonceLength: 32 octets

Le Tableau 12 décrit les caractéristiques liées au protocole et au chiffrage qui peuvent être profilées. Ces caractéristiques sont définies en détail dans l'IEC 62541-6. Il est recommandé que les *Serveurs* et les *Clients* prennent en charge le plus grand nombre possible de ces options pour une plus grande interopérabilité.

Tableau 12 – Protocole et codage

Catégorie	Intitulé	Description
Serveur	Protocole de Connexion inversée *Serveur*	Prend en charge la connectivité inversée en envoyant un message ReverseHello à un Client. La procédure de connexion inversée peut être appliquée à plusieurs transports, spécifiés dans l'IEC 62541-6, et doit être prise en charge pour tous ces transports disponibles dans un Serveur.
Serveur	Protocole Configuration	Permet l'administration des Points d'extrémité et du nombre d'accès utilisé par ces derniers.
Client	Protocole de Connexion inversée *Client*	Prend en charge la connectivité inversée en acceptant les messages ReverseHello des Serveurs et établit un Canal sécurisé si l'URI du Serveur est acceptée par le Client ou par l'utilisateur du Client. La procédure de connexion inversée peut être appliquée à plusieurs transports, spécifiés dans l'IEC 62541-6, et doit être prise en charge pour tous ces transports pris en charge par le Client.
Transport	Protocole UA TCP	Prend en charge le protocole de transport UA TCP défini dans l'IEC 62541-6.
Transport	Protocole HTTPS	Prend en charge le protocole HTTPS défini dans l'IEC 62541-6.
Transport	Protocole Web Sockets	Prend en charge le protocole WSS (WebSocket) avec au moins l'un des sous-protocoles définis dans l'IEC 62541-6.
Transport	Conversation sécurisée UA	Prend en charge la conversation sécurisée UA spécifiée dans l'IEC 62541-6.
Transport	Codage UA binaire	Prend en charge le codage UA binaire. Les valeurs de ces types de données sont codées dans des formats binaires compacts, de manière contiguë et sans balises. Autrement dit, il est admis par hypothèse que le récepteur comprend la structure qu'il décode.
Transport	Codage UA SOAP-XML	Prend en charge le codage XML SOAP V1.2 défini dans l'IEC 62541-6. Les éléments XML incluent toutes les informations nécessaires pour les reconvertir en structures OPC UA dans tous les langages.
Transport	Codage réversible JSON	Prend en charge le codage JSON réversible défini dans l'IEC 62541-6. L'élément JSON inclut toutes les informations nécessaires pour le reconvertir en structures OPC UA dans tous les langages.

5.4 Modèle d'information et caractéristiques relatives à l'AddressSpace

Le Tableau 13 décrit les éléments relatifs aux caractéristiques de base qui peuvent être intégrés dans les profils. Pour plus d'informations sur ces éléments, se reporter à l'IEC 62541-3, l'IEC 62541-5 et l'IEC 62541-6. Les *Serveurs* avec une plus grande capacité de ressources prennent en charge la plus grande part de cette fonctionnalité, un *Serveur* avec une capacité de ressources moindre ne peut prendre en charge qu'une partie de cette même fonctionnalité. De nombreux *Clients* utilisent une partie de cette fonctionnalité et des *Clients* plus robustes utilisent en revanche la plus grande partie de cette fonctionnalité.

Tableau 13 – Informations de base

Catégorie	Intitulé	Description
Serveur	Informations de base – Structure principale	Le *Serveur* prend en charge l'*Objet Serveur*, les ServerCapabilities et la structure de l'*AddressSpace* OPC UA.
Serveur	Informations de base – Capacités *Serveur*	Le *Serveur* prend en charge l'édition de la limite du *Serveur* dans les ServerCapabilities, y compris MaxArrayLength, MaxStringLength, MaxNodePerRead, MaxNodesPerWrite, MaxNodesPerSubscription et MaxNodesPerBrowse.
Serveur	Informations de base – Evénements de progression	Le *Serveur* présente si la génération d'événements de progression est prise en charge pour des appels de services de longue durée, tels que HistoryRead ou Query. Si elle est indiquée comme prise en charge dans les ServerCapabilities, alors les événements réels sont vérifiés.
Serveur	Informations de base – Diagnostics	Le *Serveur* prend en charge la collecte d'informations de diagnostic. Le champ EnabledFlag de l'*Objet* ServerDiagnostics peut être réglé sur TRUE. Dans ce cas, tous les *Objets* et *Variables* statiques et dynamiques des données de diagnostic définies dans l'IEC 62541-5 sont pris en charge.
Serveur	Informations de base – Statut Système	Le *Serveur* prend en charge la génération de SystemStatusChangeEventType indiquant l'arrêt du *Serveur* (SourceNode=*Serveur*).
Serveur	Informations de base – Temps de retour estimé	Le *Serveur* prend en charge la *Propriété* EstimatedReturnTime. Elle indique le moment où il est prévu que le ServerStatus.State du *Serveur* soit RUNNING_0. Les *Clients* peuvent utiliser cette information pour régir la logique de reconnexion.
Serveur	Informations de base – Statut Système – Système sous-jacent	Le *Serveur* prend en charge la génération de SystemStatusChangeEventType indiquant les modifications d'un Système sous-jacent (SourceNode = *Serveur*). Cet événement peut aussi être utilisé pour indiquer que le *Serveur* OPC UA a des systèmes sous-jacents.
Serveur	Informations de base – Défaillance d'appareil	Le *Serveur* prend en charge la génération des DeviceFailureEventType indiquant les modifications apportées à chaque appareil dans un système sous-jacent.
Serveur	Informations de base – *Méthode* GetMonitoredItems	Le *Serveur* prend en charge l'obtention d'informations relatives aux abonnements via la *Méthode* GetMonitoredItems sur l'objet *Serveur*.
Serveur	Informations de base – *Méthode* ResendData	Prend en charge la *Méthode* normalisée ResendData (définie dans l'IEC 62541-5) pour obtenir la dernière valeur des éléments surveillés d'un *Abonnement*.
Serveur	Informations de base – Système type	Le *Serveur* présente un Système Type avec des DataTypes, des ReferenceTypes, des *ObjectTypes* et des VariableTypes, y compris tous les types OPC UA (namespace 0) que le *Serveur* utilise, selon la définition de l'IEC 62541-5. Les éléments définis dans Namespace 0, mais définis dans d'autres parties de la spécification sont soumis à l'essai dans le cadre des autres modèles d'information.
Serveur	Informations de base – Système type personnalisé	Le *Serveur* prend en charge les types personnalisés (c'est-à-dire les types dérivés des *ObjectTypes*, VariableTypes, ReferenceTypes ou DataTypes connus). La prise en charge de cette unité de conformité exige que les types personnalisés et leur arbre d'héritage complet soient présentés dans l'*AddressSpace*.

Catégorie	Intitulé	Description
Serveur	Informations de base – Modification de modèle	Le *Serveur* prend en charge l'*Evénement* ModelChange et la *Propriété* NodeVersion pour tous les *Nœuds* pour lesquels le serveur permet des modifications du Modèle.
Serveur	Informations de base – Règles de modélisation – Paramètre fictif	Le *Serveur* prend en charge la définition d'*Objet* personnalisé ou de *Variables* qui comprennent l'utilisation des règles de modélisation OptionalPlaceholder ou MandatoryPlaceholder.
Serveur	Informations de base – SemanticChange	Le *Serveur* prend en charge SemanticChangeEvent pour certaines Propriétés. Cela comprend le réglage du bit SemanticChange dans le statut lorsqu'un changement sémantique se produit, comme une modification de l'unité technique associée à une valeur.
Serveur	Informations de base – EventType EventQueueOverflow	Le *Serveur* prend en charge l'EventQueueOverflowEventType défini dans l'IEC 62541-4.
Serveur	Informations de base – OptionSet	Le *Serveur* prend en charge le *VariableType* OptionSet.
Serveur	Informations de base – ValueAsText	Le *Serveur* prend en charge la *Propriété* ValueAsText pour les DataTypes répertoriés.
Serveur	Informations de base – Unités techniques	Le *Serveur* prend en charge la définition des *Variables* comprenant la *Propriété* Unités techniques. Cette propriété utilise la structure de données EUInformation. Cette structure par défaut représente les "Codes for Units of Measurement" de l'UN/CEFACT. Si une autre représentation EU est exigée, l'EUInformation.namespaceUri indique l'espace de noms alternatif.
Serveur	Informations de base – Liste de sélection	Le *Serveur* prend en charge les Variables du VariableType SelectionListType.
Serveur	Informations de base – FileType Base	Le *Serveur* prend en charge l'*Objet* FileType (voir l'IEC 62541-5). L'écriture de fichier peut être limitée.
Serveur	Informations de base – FileType Ecriture	Le *Serveur* prend en charge l'*Objet* FileType, y compris l'écriture des fichiers. La prise en charge du contrôle d'accès des utilisateurs dans l'*Objet* FileType est également incluse.
Serveur	Informations de base – Méthode RequestServerStateChange	Le *Serveur* prend en charge la *Méthode* RequestServerStateChange.
Serveur	Informations de base – Instance de diagramme d'états	Prend en charge les instances de StateMachineType ou un sous-type dans l'*AddressSpace*. Génère des Evénements lorsque des changements d'état significatifs ont lieu. Il existe au moins une *Référence* GeneratesEvent pour définir un ou plusieurs *Evénement*(s) déclenché(s) aux changements d'état.
Serveur	Informations de base – Instance de diagramme d'états finis	Prend en charge les instances de FiniteStateMachineType ou un sous-type dans l'*AddressSpace*.
Serveur	Informations de base – Etats et transitions disponibles	Prend en charge les Propriétés AvailableStates et AvailableTransitions pour le FiniteStateMachineType.
Client	Informations de base *Client* – Base	Le Client utilise l'*AddressSpace* OPC UA défini. Accède ou permet l'accès aux informations du *Serveur* telles que l'état du *Serveur*, BuildInfo, les capacités, le tableau d'espace de noms et le modèle de type.
Client	Informations de base *Client* – Respecter les limites de fonctionnement	Le Client doit respecter les limites du *Serveur* décrites dans l'*Objet* ServerCapabilites du *Serveur*.
Client	Informations de base *Evénements* – Traitement	Le *Client* est capable de s'abonner aux Événements OPC UA de base et de les traiter.
Client	Informations de base *Client* – Statut de système	Le *Client* utilise SystemStatusChangeEventType pour détecter un arrêt du Serveur.
Client	Informations de base *Client* – Temps de retour estimé	Le *Client* utilise la *Propriété* EstimatedReturnTime pour régir la logique de reconnexion.
Client	Informations de base *Client* – Statut système – Système sous-jacent	Le *Client* utilise le SystemStatusChangeEventType pour détecter les modifications d'un Système Sous-jacent (SourceNode = *Serveur*).

Catégorie	Intitulé	Description
Client	Informations de base *Client* – Défaillance d'appareil	Le *Client* utilise le DeviceFailureEventType pour détecter les appareils défaillants dans des systèmes sous-jacents.
Client	Informations de base *Client* – Evénements de progression	Le *Client* utilise ProgressEvents, y compris la vérification de leur prise en charge.
Client	Informations de base *Client* – Diagnostics	Le *Client* permet un accès interactif ou par programmation aux informations de diagnostic du *Serveur*.
Client	Informations de base Client – Programmation type	Le *Client* traite par programmation les instances d'*Objets* ou de Variables en se servant de leurs définitions de type. Ceci inclut les DataTypes, les *ObjectTypes* et les VariableTypes.
Client	Informations de base *Client* – Connaissances préalables types	Le *Client* doit interagir avec les *Serveurs* qui ne présentent pas les types d'OPC UA dans l'*AddressSpace*.
Client	Informations de base *Client* – *Nœuds* distants	Le *Client* peut accéder aux Nœuds ayant un NodeID étendu qui référence un *Serveur* différent du *Serveur* d'origine. Il est acceptable que les informations de configuration du *Serveur* soient préconfigurées sur le *Client*.
Client	Informations de base *Client* – Evénements de modification	Le *Client* traite les ModelChangeEvents afin de détecter les modifications de l'*AddressSpace* OPC UA du *Serveur* et de prendre les mesures appropriées pour une modification donnée.
Client	Informations de base *Client* – *Méthode* GetMonitoredItems	Le *Client* utilise la *Méthode* GetMonitoredItems pour reprendre après des interruptions de communication et/ou pour récupérer des informations relatives aux abonnements.
Client	Données de base *Client* – *Méthode* ResendData	Le *Client* utilise la *Méthode* ResendData pour rechercher la dernière valeur des éléments surveillés par les données.
Client	Informations de base *Client* – Liste de sélection	Le *Client* utilise et comprend les Variables du VariableType SelectionListType.
Client	Informations de base *Client* – FileType Base	Le *Client* peut accéder à un *Objet* FileType pour transférer un fichier du *Serveur* au *Client*. Cela comprend des fichiers de grande taille.
Client	Informations de base *Client* – FileType Ecriture	Le *Client* peut accéder à un *Objet* FileType pour transférer un fichier du *Client* au *Serveur*. Cela comprend des fichiers de grande taille.
Client	Informations de base *Client* – RequestServerStateChange	Le *Client* peut invoquer la *Méthode* RequestServerStateChange.
Client	Informations de base *Client* – Instance de diagramme d'états	Utilise les instances de StateMachineType ou un sous-type. Surveille la *Variable* du composant CurrentState de l'instance ou les Evénements déclenchés sous l'effet des changements d'état. Utilise les Méthodes, si elles sont définies, pour que le StateMachineType affecte l'état.
Client	Informations de base *Client* – Instance de diagramme d'états finis	Utilise les instances de FinitStateMachineType ou d'un sous-type. Surveille la *Variable* du composant CurrentState de l'instance ou les Evénements déclenchés sous l'effet des changements d'état.
Client	Informations de base *Client* – Etats et transitions disponibles	Utilise les Propriétés AvailableStates et AvailableTransitions lorsqu'elles sont présentées par un *Serveur*.

Le Tableau 14 décrit les éléments relatifs aux informations concernant le modèle d'espace d'adressage qui peuvent être intégrés dans les profils. Les détails de ces éléments de modèle sont définis dans l'IEC 62541-3 et l'IEC 62541-5. Cela inclut les *Facettes serveur* qui décrivent les éléments présentés par un *Serveur* et les *Facettes client* qui décrivent les éléments consommés par un *Client*.

Tableau 14 – Modèle d'espace d'adressage

Catégorie	Intitulé	Description
Serveur	Espace d'adressage – Base	Prend en charge les *NodeClasses* avec leurs *Attributs* et leurs *Références*, définis dans l'IEC 62541-3. Ceci inclut, par exemple, les attributs suivants: *Objet, ObjectType, Variable, VariableType, Références* et *DataType.*
Serveur	Espace d'adressage – Atomicité	Prend en charge le réglage des fanions NonatomicRead et NonatomicWrite dans l'*Attribut* AccessLevelEx pour que les *Nœuds de Variables* indiquent si les opérations de lecture ou d'écriture peuvent être réalisées de manière atomique. Si les fanions sont réglés sur '1', l'atomicité ne peut pas être assurée.
Serveur	Espace d'adressage – Matrice complète seulement	Prend en charge le réglage du fanion WriteFullArrayOnly dans l'*Attribut* AccessLevelEx pour que les *Nœuds de Variables* des types de données non scalaires indiquent si les opérations d'écriture peuvent être réalisées avec une plage d'indices pour une matrice.
Serveur	Espace d'adressage – Evénements	Prend en charge les éléments de l'*AddressSpace* OPC UA afin de générer les notifications d'*Evénements*. Ceci inclut au moins un *Nœud* avec un *Attribut EventNotifier* mis à True (*Nœud Serveur*).
Serveur	Espace d'adressage – Dictionnaire de données complexes	Prend en charge les DataTypes structurés avec un Dictionnaire de Données. Noter que la V1.04 de l'IEC 62541-3 spécifie une approche simplifiée utilisant le nouvel Attribut DataTypeDefinition. L'unité de conformité "Address Space DataTypeDefinition Attribute" exige la prise en charge de l'Attribut DataTypeDefinition. La prise en charge d'un DataDictionary est déconseillée dans l'une des versions ultérieures de l'OPC UA.
Serveur	Espace d'adressage – Attribut DataTypeDefinition	Prend en charge les DataTypes structurés et présente les métadonnées et les informations sur le codage avec un StructureDefinitionType au moyen de l'*Attribut* DataTypeDefinition.
Serveur	Espace d'adressage – *Méthode*	Prend en charge les *Nœuds* de *Méthode*.
Serveur	Espace d'adressage – Hiérarchie de Notification	Prend en charge l'utilisation de la référence HasNotifier afin de construire une hiérarchie des *Nœuds d'Objet* qui sont des notifications avec d'autres *Nœuds* d'*Objet* de notification.
Serveur	Espace d'adressage – Hiérarchie de Source	Prend en charge des hiérarchies de sources d'événements dont chacune prend racine dans un *Nœud* d'*Objet* qui est une notification. La *Référence* HasEventSource est utilisée pour relier les *Nœuds* dans une hiérarchie. Si des *Conditions* sont prises en charge, la hiérarchie doit comprendre les *Références* HasCondition.
Serveur	Espace d'adressage – WriteMask	Prend en charge le WriteMask qui indique la disponibilité de l'accès en écriture pour tous les attributs, y compris les attributs non pris en charge.
Serveur	Espace d'adressage – UserWriteMask	Prend en charge le UserWriteMask qui indique la disponibilité de l'accès en écriture pour tous les attributs d'un utilisateur donné, y compris les attributs non pris en charge. La prise en charge comprend au moins deux niveaux d'utilisateurs.
Serveur	Espace d'adressage – UserWriteMask Multiniveaux	Prend en charge le UserWriteMask qui indique la disponibilité de l'accès en écriture pour tous les attributs d'un utilisateur donné, y compris les attributs non pris en charge. La prise en charge comprend plusieurs niveaux de contrôle de l'accès pour tous les nœuds du système.
Serveur	Espace d'adressage – Niveau d'accès utilisateur complet	Met en œuvre la sécurité du niveau d'accès de l'utilisateur; cela comprend la prise en charge de plusieurs niveaux de contrôle de l'accès pour les nœuds de *Variables* dans le système. Cela comprend une indication d'accès à l'*Attribut* Valeur en lecture, écriture, lecture Historique et écriture Historique.
Serveur	Espace d'adressage – Niveau d'accès utilisateur de base	Met en œuvre la sécurité du niveau d'accès de l'utilisateur pour les nœuds de *Variables*; cela comprend au moins deux utilisateurs dans le système. Cela comprend une indication d'accès à l'Attribut Valeur en lecture, écriture, lecture Historique et écriture Historique.
Client	Espace d'adressage *Client* – Base	Utilise et comprend les *NodeClasses* avec leurs *Attributs* et leur comportement, définis dans l'IEC 62541-3. Ceci inclut, par exemple, les attributs suivants: *Objet, ObjectType, Variable, VariableType, Références* et *DataType.* Ceci inclut de traiter les BrowseNames et NodeIds de Chaîne comme sensibles à la casse.

Catégorie	Intitulé	Description
Client	Espace d'adressage *Client* – Atomicité	Accède aux fanions NonatomicRead ou NonatomicWrite dans l'*Attribut* AccessLevelEx des *Nœuds de Variables* pour définir si les opérations de lecture ou d'écriture peuvent être réalisées de manière atomique. Ces informations s'affichent normalement à l'attention de l'utilisateur aux fins d'actions ultérieures.
Client	Espace d'adressage *Client* – Matrice complète seulement	Accède au fanion WriteFullArrayOnly dans l'*Attribut* AccessLevelEx des *Nœuds de Variables* ayant des types de données non scalaires pour déterminer si les opérations d'écriture avec une IndexRange sont autorisées dans une matrice.
Client	Espace d'adressage *Client* – Dictionnaire de données complexes	Utilise et comprend les DataTypes structurés arbitraires au moyen du Dictionnaire de Données. Noter que la V1.04 de l'IEC 62541-3 spécifie une approche simplifiée utilisant le nouvel Attribut DataTypeDefinition. L'unité de conformité "Address Space Client DataTypeDefinition Attribute" exige la prise en charge de l'Attribut DataTypeDefinition.
Client	Espace d'adressage *Client* – Attribut DataTypeDefinition	Utilise et comprend les DataTypes structurés arbitraires où les métadonnées et les informations sur le codage sont présentées avec un StructureDefinitionType au moyen de l'*Attribut* DataTypeDefinition.
Client	Espace d'adressage *Client* – Hiérarchie de Notification	Utilise la hiérarchie des *Nœuds* d'*Objet* qui sont des notifications afin de détecter des zones spécifiques où le *Client* peut s'abonner à des Evénements.
Client	Espace d'adressage *Client* – Hiérarchie de Source	Détecte et utilise la hiérarchie des sources d'événements présentée pour des *Nœuds* d'*Objet* spécifiques qui sont des notifications d'événements.

Le Tableau 15 décrit les éléments relatifs au modèle d'information d'accès aux données qui peuvent être intégrés dans les profils. Les détails de ce modèle sont définis dans l'IEC 62541-8. Les *Serveurs* peuvent présenter ce modèle d'information et les *Clients* peuvent utiliser ce modèle d'information.

Tableau 15 – Accès aux données

Catégorie	Intitulé	Description
Serveur	Accès aux données – DataItems	Fournit des *Variables* du DataItemType ou un de ses sous-types. Prend en charge les StatusCodes spécifiés dans l'IEC 62541-8. La prise en charge des propriétés facultatives ("InstrumentRange", par exemple) doit être vérifiée lors des essais de certification et est indiquée sur le *Certificat*.
Serveur	Accès aux données – AnalogItems	Prend en charge les *Variables* AnalogItemType avec les Propriétés correspondantes. La prise en charge des propriétés facultatives est indiquée.
Serveur	Accès aux données – PercentDeadband	Prend en charge le filtre PercentDeadband lors de la surveillance des *Variables* AnalogItemType.
Serveur	Accès aux données – SemanticChanges	Prend en charge les changements sémantiques des éléments AnalogItemType (*Propriété* EURange et/ou *Propriété* EngineeringUnits). Prend en charge les bits de StatusCode des changements sémantiques, le cas échéant.
Serveur	Accès aux données – TwoState	Prend en charge les *Variables* TwoStateDiscreteType avec les Propriétés correspondantes.
Serveur	Accès aux données – MultiState	Prend en charge les *Variables* MultiStateDiscreteType avec les Propriétés correspondantes.
Serveur	Accès aux données – MultiStateValueDiscrete	Prend en charge les *Variables* MultiStateValueDiscreteType avec les Propriétés correspondantes.
Serveur	Accès aux données – ArrayItemType	Fournit des *Variables* de l'ArrayItemType ou l'un de ses sous-types (YArrayItemType, XYArrayItemType, ImageArrayType, CubeArrayType et NDimensionArrayType). Les sous-types pris en charge sont répertoriés. La prise en charge de ce type comprend la prise en charge de toutes les propriétés obligatoires, y compris AxisInformation.
Serveur	Accès aux données – Nombre Complexe	Prend en charge le type de données de Nombre Complexe. Ce type de données est disponible pour tout type de variable n'ayant pas d'autre restriction explicite.
Serveur	Accès aux données – Nombre DoubleComplex	Prend en charge le type de données de Nombre DoubleComplex. Ce type de données est disponible pour tout type de variable n'ayant pas d'autre restriction explicite.
Client	Accès aux données *Client* – Base	Comprend les types de *Variables* DataAccess. Utilise les Propriétés normalisées, le cas échéant.
Client	Accès aux données *Client* – AnalogItems	Comprend les *Variables* AnalogItemType avec les Propriétés correspondantes.
Client	Accès aux données *Client* – TwoState	Comprend les *Variables* TwoStateDiscreteType avec les Propriétés correspondantes.
Client	Accès aux données *Client* – MultiState	Comprend les *Variables* MultiStateDiscreteType avec les Propriétés correspondantes.
Client	Accès aux données *Client* – MultiStateValueDiscrete	Comprend les *Variables* MultiStateValueDiscrete avec les Propriétés correspondantes.
Client	Accès aux données *Client* – Bande Morte	Utilise PercentDeadband pour filtrer les modifications de valeurs des *Variables* AnalogItemType.
Client	Accès aux données *Client* – SemanticChange	Reconnaît le bit de changement sémantique dans le StatusCode tout en surveillant les éléments et en prenant les mesures appropriées. Généralement, le *Client* doit relire les Propriétés qui définissent la sémantique spécifique au type, telles que les propriétés EURange et EngineeringUnits.

Le Tableau 16 décrit les éléments relatifs au modèle d'information *Alarme* et *Conditions* qui peuvent être intégrés dans les profils. Les détails de ce modèle sont définis dans l'IEC 62541-9. Les *Serveurs* qui traitent des *Alarmes* et *Conditions* présentent ce modèle d'information et les *Clients* qui traitent des *Alarmes* et *Conditions* utilisent ce même modèle.

Tableau 16 – Alarmes et conditions

Catégorie	Intitulé	Description
Serveur	A & C – Base	Prend en charge le ConditionType du modèle d'*Alarme & Condition*.
Serveur	A & C – Activer	Prend en charge les Méthodes Activer et Désactiver.
Serveur	A & C – Rafraîchissement	Prend en charge la *Méthode* ConditionRefresh et le concept de rafraîchissement
Serveur	A & C – Refresh2	Prend en charge la *Méthode* ConditionRefresh2 et le concept de rafraîchissement fondé sur un élément surveillé.
Serveur	A & C – Instances	Prend en charge la présentation des instances de *Conditions A&C* dans l'*AddressSpace*.
Serveur	A & C – ConditionClasses	Prend en charge les différentes classes de *Conditions* pour le regroupement et le filtrage des *Alarmes*.
Serveur	A & C – Sous Classes de *Conditions*	Prend en charge l'assignation des différentes sous-classes de *Conditions* pour le regroupement et le filtrage des *Alarmes*.
Serveur	A & C – Acquittement	Prend en charge le concept d'Aquittement, la *Méthode* d'Aquittement et le Type AcknowledgeableCondition.
Serveur	A & C – Confirmer	Prend en charge le concept de confirmation et la *Méthode* Confirmer.
Serveur	A & C – Commentaire	Prend en charge le concept de Commentaires et la *Méthode* AddComments.
Serveur	A & C – *Alarme*	Prend en charge les caractéristiques obligatoires du Type AlarmCondition.
Serveur	A & C – Paramètres d'*Alarme*	Prend en charge la collecte des données des paramètres d'alarme définies dans l'IEC 62541-9. Cela implique une instance parmi d'autres de l'AlarmMetricsType.
Serveur	A & C – Branche	Prend en charge l'embranchement des Types de *Conditions* et tous les sous-types, par exemple AcknowledgeableConditionType et AlarmConditionType, etc.
Serveur	A & C – Suspension	Prend en charge le concept de suspension et les méthodes TimedShelve, OneShotShelve et Unshelve.
Serveur	A & C – Suppression	Prend en charge le SuppressedState.
Serveur	A & C – Suppression par l'opérateur	Prend en charge les *Méthodes* Supprimer et UnSuppress pour permettre à un opérateur d'avoir un contrôle sur le SuppressedState.
Serveur	A & C – Silence	Prend en charge le concept de silence et la *Méthode* Silence.
Serveur	A & C – Hors Service	Prend en charge l'état OutOfService et la *Méthode* OutOfService.
Serveur	A & C – Activation et désactivation différées	Prend en charge les Propriétés OnDelay et OffDelay pour éliminer les *Alarmes* perturbatrices.
Serveur	A & C – Répétition des alarmes	Prend en charge les Propriétés ReAlarmTime et ReAlarmRepeatCount qui définissent la répétition automatique des *Alarmes* dans certaines conditions.
Serveur	A & C – Alarme "First in Group"	Prend en charge les éléments "FirstInGroup" d'une *Alarme*, qui indiquent quelle *Alarme* au sein d'un groupe a déclenché les autres.
Serveur	A & C – Son Audible	Prend en charge la *Propriété* AudibleSound. Cette *Propriété* contient le fichier son à lire si une *Alarme* sonore est générée.
Serveur	A & C – Niveau exclusif	Prend en charge le type d'*Alarme* de Niveau Exclusif.
Serveur	A & C – Limite exclusive	Prend en charge les *Alarmes* de Limite exclusive. Il importe qu'un *Serveur* qui prend en charge ce type d'alarme prenne en charge au moins l'un des sous-types: Niveau, Ecart ou RateofChange.
Serveur	A & C – Ecart exclusif	Prend en charge le type d'*Alarme* d'Ecart exclusif.
Serveur	A & C – RateOfChange exclusif	Prend en charge le type d'*Alarme* de RateOfChange exclusif.
Serveur	A & C – Limite non exclusive	Prend en charge les *Alarmes* de Limite Non Exclusive. Il importe qu'un *Serveur* qui prend en charge ce type d'alarme prenne en charge au moins l'un des sous-types: Niveau, Ecart ou RateofChange.
Serveur	A & C – Niveau non exclusif	Prend en charge le type d'*Alarme* de Niveau Non Exclusif.
Serveur	A & C – Ecart non exclusif	Prend en charge le type d'*Alarme* d'Ecart Non Exclusif.

Catégorie	Intitulé	Description
Serveur	A & C – RateOfChange non exclusif	Prend en charge le type d'*Alarme* de RateOfChange Non Exclusif.
Serveur	A & C – Discrètes	Prend en charge les types d'*Alarmes* discrètes.
Serveur	A & C – OffNormal	Prend en charge l'OffNormalAlarmType.
Serveur	A & C – OffNormal	Prend en charge SystemOffNormalAlarmType.
Serveur	A & C – Déclenchement	Prend en charge le type d'*Alarme* de Déclenchement.
Serveur	A & C – Anomalie	Prend en charge le type d'*Alarme* d'Anomalie.
Serveur	A & C – Dialogue	Prend en charge le DialogConditionType, y compris la *Méthode* Répondre.
Serveur	A & C – CertificateExpiration	Prend en charge le CertificateExpirationAlarmType.
Serveur	A & E – Conteneur de Mapping	Le *Serveur* utilise le mapping COM A&E spécifié dans l'annexe de l'IEC 62541-9 pour mettre en correspondance les Evénements OPC-COM et les Evénements A&C. Ceci inclut le mapping de la Classe *Condition*.
Client	A & C *Client* – Base	Utilise le ConditionType du modèle d'*Alarme & Condition*.
Client	A & C *Client* – Activer	Utilise les Méthodes Activer et Désactiver.
Client	A & C *Client* – Rafraîchissement	Utilise la *Méthode* ConditionRefresh et le concept de rafraîchissement.
Client	A & C *Client* – Refresh2	Utilise la *Méthode* ConditionRefresh2 et le concept de rafraîchissement fondé sur un élément surveillé.
Client	A & C *Client* – Instances	Utilise les instances de *Conditions* A&C lorsqu'elles sont présentées dans l'*AddressSpace*.
Client	A & C *Client* – ConditionClasses	Utilise les classes de *Conditions* pour grouper les *Alarmes*.
Client	A & C *Client* – Sous Classes de *Conditions*	Utilise les sous-classes de *Conditions* pour grouper ou pour filtrer les *Alarmes*.
Client	A & C *Client* – Acquittement	Comprend le concept d'Acquittement et le Type AcknowledgeableCondition, et utilise la *Méthode* d'Acquittement si elle est demandée.
Client	A & C *Client* – Confirmer	Comprend le concept de confirmation des *Conditions* et utilise la *Méthode* Confirmer.
Client	A & C *Client* – Commentaire	Comprend le concept de Commentaires et la *Méthode* AddComments.
Client	A & C *Client* – Alarme	Comprend le concept d'*Alarmes* et utilise les caractéristiques obligatoires du Type AlarmCondition.
Client	A & C *Client* – Paramètres d'*Alarme*	Comprend et utilise les données des Paramètres d'*Alarme* définis dans l'IEC 62541-9. Cela implique la découverte d'instances de l'AlarmMetricsType, qui peuvent exister partout dans la hiérarchie HasNotifier.
Client	A & C *Client* – Branche	Peut utiliser et traiter les Branches de *Conditions*, y compris toutes les actions associées aux instances de *Conditions* précédentes.
Client	A & C *Client* – Suspension	Utilise le modèle de suspension, y compris les Méthodes TimedShelve, OneShotShelve et Unshelve.
Client	A & C *Client* – Suppression	Comprend le modèle de SuppressedState.
Client	A & C *Client* – Suppression par l'opérateur	Utilise les *Méthodes* Supprimer et UnSuppress pour permettre à un opérateur d'avoir un contrôle sur le SuppressedState.
Client	A & C *Client* – Silence	Comprend le modèle SilencedState et utilise la Méthode Silence.
Client	A & C *Client* – Hors Service	Comprend le modèle OutOfServiceState et utilise la Méthode OutOfService.
Client	A & C *Client* – Activation et désactivation différées	Utilise les Propriétés OnDelay et OffDelay pour éliminer les *Alarmes* perturbatrices.

Catégorie	Intitulé	Description
Client	A & C *Client* – Répétition des alarmes	Comprend et utilise les Propriétés ReAlarmTime et ReAlarmRepeatCount. Configure la *Propriété* ReAlarmTime pour la répétition automatique d'une *Alarme*. Noter que la configuration est possible seulement pour les *Serveurs* qui présentent des instances d'*Alarme*.
Client	A & C *Client* – *Alarme* "First in Group"	Utilise les éléments "FirstInGroup" d'une *Alarme* pour déterminer quelle *Alarme* au sein d'un groupe a déclenché les autres.
Client	A & C *Client* – Son Audible	Utilise la *Propriété* AudibleSound et lit le fichier son (le cas échéant).
Client	A & C *Client* – Niveau exclusif	Utilise les *Alarmes* de Niveau Exclusif.
Client	A & C *Client* – Limite exclusive	Utilise les *Alarmes* de Limite Exclusive. Exige l'utilisation d'au moins un des sous-types.
Client	A & C *Client* – Ecart exclusif	Utilise les *Alarmes* d'Ecart Exclusif.
Client	A & C *Client* – RateOfChange exclusif	Utilise les *Alarmes* de RateOfChange exclusif.
Client	A & C *Client* – Niveau non exclusif	Utilise les *Alarmes* de Niveau Non Exclusif
Client	A & C *Client* – Limite non exclusive	Utilise les *Alarmes* de Limite Non Exclusive. Exige l'utilisation d'au moins un des sous-types.
Client	A & C *Client* – Ecart non exclusif	Utilise les *Alarmes* d'Ecart Non Exclusif.
Client	A & C *Client* – RateOfChange non exclusif	Utilise les *Alarmes* de RateOfChange non exclusif.
Client	A & C *Client* – Discrètes	Utilise les types d'*Alarmes* discrètes.
Client	A & C *Client* – OffNormal	Utilise l'OffNormalAlarmType.
Client	A & C *Client* – SystemOffNormal	Utilise le SystemOffNormalAlarmType.
Client	A & C *Client* – Déclenchement	Utilise le TripAlarmType.
Client	A & C *Client* – Anomalie	Utilise le type d'*Alarme* d'Anomalie.
Client	A & C *Client* – Dialogue	Utilise DialogConditionType, y compris la *Méthode* Répondre.
Client	A & C *Client* – CertificateExpiration	Utilise le CertificateExpirationAlarmType.

Le Tableau 17 décrit les éléments relatifs au modèle d'information d'accès aux données historiques qui peuvent être intégrés dans les profils. Les détails de ce modèle sont définis dans l'IEC 62541-11. Les *Serveurs* qui prennent en charge un niveau de données historiques présentent ce modèle d'information et les *Clients* qui utilisent les données historiques utilisent ce même modèle.

Tableau 17 – Accès à l'historique

Catégorie	Intitulé	Description
Serveur	Accès à l'historique – Lecture de Données Brutes	Fournit une prise en charge générale de l'Accès à l'historique de base, avec la lecture de données brutes en utilisant la structure ReadRawModifiedDetails. Si l'intervalle de temps est spécifié avec un temps de départ et d'arrêt ainsi qu'un nombre de valeurs (il importe de fournir au moins deux des trois paramètres), le fanion ReadModified est mis à FALSE.
Serveur	Accès à l'historique – Point de Continuation de Lecture des Données avec un Maximum de *Nœuds*	Prend en charge suffisamment de points de continuation pour couvrir le nombre de points pris en charge indiqué dans la *Propriété* OperationLimits du *Serveur* MaxNodesPerHistoryReadData pour accéder aux données historiques.
Serveur	Accès à l'historique – Instance Temporelle	Prend en charge la lecture des données historiques à une instance spécifiée dans le temps à l'aide de la structure ReadAtTimeDetails.
Serveur	Accès à l'historique – Agrégats	Prend en charge la lecture d'un ou plusieurs Agrégats de valeurs historiques de *Variables* à l'aide de la structure ReadProcessedDetails. Il importe de prendre en charge au moins l'un des Agrégats décrits dans l'IEC 62541-13.
Serveur	Accès à l'historique – Insertion de Valeurs	Prend en charge l'insertion de valeurs historiques de *Variables*.
Serveur	Accès à l'historique – Suppression de Valeurs	Prend en charge la suppression de valeurs historiques de *Variables*.
Serveur	Accès à l'historique – Mise à Jour de Valeurs	Prend en charge la mise à jour de valeurs historiques de *Variables*.
Serveur	Accès à l'historique – Remplacement de Valeurs	Prend en charge le remplacement de valeurs historiques de *Variables*.
Serveur	Accès à l'historique – Valeurs Modifiées	Prend en charge le maintien d'anciennes valeurs pour les données historiques qui ont été mises à jour et la récupération de ces valeurs à l'aide de la structure ReadRawModifiedDetails (fanion ReadModified mis à TRUE).
Serveur	Accès à l'historique – Annotations	Prend en charge la saisie et la récupération des Annotations pour les données historiques. La récupération est effectuée à l'aide de la fonctionnalité de lecture de données historiques brutes normalisée (ReadRawModifiedDetails). La saisie utilise la fonctionnalité de mise à jour historique (UpdateStructureDataDetails) normalisée.
Serveur	Accès à l'historique – ServerTimestamp	Prend en charge l'apport d'un ServerTimestamp (ainsi que le SourceTimestamp par défaut).
Serveur	Accès à l'historique – Lecture des données structurées brutes	Prend en charge l'Accès à l'historique ReadRawModified pour les données structurées. La prise en charge de la structure pour une annotation n'est pas jugée comme une prise en charge des données structurées génériques.
Serveur	Accès à l'historique – Instance temporelle de données structurées	Prend en charge l'Accès à l'historique pour les données structurées. Prend en charge ReadAtTimeDetails pour les données structurées. La prise en charge de la structure pour une annotation n'est pas jugée comme une prise en charge des données structurées génériques.
Serveur	Accès à l'historique – Insérer des données structurées	Prend en charge l'Accès à l'historique pour les données structurées. Insère des données structurées. La prise en charge de la structure pour une annotation n'est pas jugée comme une prise en charge des données structurées génériques.
Serveur	Accès à l'historique – Supprimer des données structurées	Prend en charge l'Accès à l'historique pour les données structurées. Supprime les données existantes. La prise en charge de la structure pour une annotation n'est pas jugée comme une prise en charge des données structurées génériques.
Serveur	Accès à l'historique – Mise à Jour de données structurées	Prend en charge l'Accès à l'historique pour les données structurées. Met à jour les données existantes. La prise en charge de la structure pour une annotation n'est pas jugée comme une prise en charge des données structurées génériques.
Serveur	Accès à l'historique – Remplacer des données structurées	Prend en charge le remplacement de données historiques structurées. La prise en charge de la structure pour une annotation n'est pas jugée comme une prise en charge des données structurées génériques.

Catégorie	Intitulé	Description
Serveur	Accès à l'historique – Lecture de données structurées modifiées	Prend en charge le maintien d'anciennes valeurs pour les données historiques structurées qui ont été mises à jour et la récupération de ces valeurs. Utilise la structure ReadRawModifiedDetails (fanion ReadModified mis à TRUE) pour les données structurées. La prise en charge de la structure pour une annotation n'est pas jugée comme une prise en charge des données structurées génériques.
Serveur	Accès à l'historique – Evénements	Prend en charge la récupération d'Evénements historiques à l'aide de la structure ReadEventDetails. Cela comprend la prise en charge du filtrage simple d'Evénements. Les champs d'*Evénements* stockés sont spécifiques au serveur, mais au moins les champs obligatoires de baseEventType sont exigés.
Serveur	Accès à l'historique – Evénements – Point de Continuation de Lecture d'*Evénements* Max	Prend en charge suffisamment de points de continuation pour couvrir le nombre de lectures d'*Evénements* prises en charge indiqué dans la *Propriété* OperationLimits du *Serveur* MaxNodesPerHistoryReadEvents pour l'Accès à l'historique d'*Evénements*.
Serveur	Accès à l'historique – Insertion d'*Evénements*	Prend en charge l'insertion d'Evénements historiques.
Serveur	Accès à l'historique – Mise à Jour d'*Evénements*	Prend en charge la mise à jour d'Evénements historiques.
Serveur	Accès à l'historique – Remplacement d'*Evénements*	Prend en charge le remplacement d'Evénements historiques.
Serveur	Accès à l'historique – Suppression d'*Evénements*	Prend en charge la suppression d'Evénements historiques.
Client	Accès à l'historique *Client* – Parcourir	Utilise le *Jeu de services* Vue pour découvrir des *Nœuds* avec des données historiques.
Client	Accès à l'historique *Client* – Lecture de Données Brutes	Utilise le *Service* HistoryRead pour lire des données historiques brutes à l'aide de la Structure ReadRawModifiedDetails (fanion ReadModified mis à FALSE).
Client	Accès à l'historique *Client* – Lecture de Données Modifiées	Utilise le *Service* HistoryRead pour lire des données historiques modifiées à l'aide de la Structure ReadRawModifiedDetails (fanion ReadModified mis à TRUE).
Client	Accès à l'historique *Client* – Lecture d'Agrégats	Utilise le *Service* HistoryRead pour lire des données historiques Agrégées. Cela comprend l'utilisation d'au moins un des Agrégats définis dans l'IEC 62541-13.
Client	Accès à l'historique *Client* – Données brutes structurées	Utilise le *Service* HistoryRead pour lire des données historiques brutes à l'aide de la Structure ReadRawModifiedDetails (fanion ReadModified mis à FALSE) pour les données structurées.
Client	Accès à l'historique *Client* – Lecture de données structurées modifiées	Utilise le *Service* HistoryRead pour lire des données historiques structurées modifiées à l'aide de la Structure ReadRawModifiedDetails (fanion ReadModified mis à TRUE).
Client	Accès à l'historique *Client* – Insérer des données structurées	Utilise le *Service* HistoryUpdate pour insérer des valeurs de données historiques pour les données structurées.
Client	Accès à l'historique *Client* – Supprimer des données structurées	Utilise le *Service* HistoryUpdate pour supprimer des valeurs de données historiques pour les données structurées.
Client	Accès à l'historique *Client* – Mettre à Jour des données structurées	Utilise le *Service* HistoryUpdate pour mettre à jour des valeurs de données historiques pour les données structurées.
Client	Accès à l'historique *Client* – Remplacer des données structurées	Utilise le *Service H*istoryUpdate pour remplacer des valeurs de données historiques pour les données structurées.
Client	Accès à l'historique *Client* – Instance temporelle de données structurées	Lit les données historiques à une instance spécifiée dans le temps pour les données structurées. Utilise la structure ReadAtTimeDetails.

Catégorie	Intitulé	Description
Client	Accès à l'historique *Client* – Lecture d'*Evénements*	Utilise le *Service* HistoryRead pour lire les données historiques d'*Evénements* à l'aide de la Structure ReadEventDetails.
Client	Accès à l'historique *Client* – Insertion d'*Evénements*	Utilise le *Service* HistoryUpdate pour insérer des Evénements historiques.
Client	Accès à l'historique *Client* – Mise à Jour d'*Evénements*	Utilise le *Service* HistoryUpdate pour mettre à jour des Evénements historiques.
Client	Accès à l'historique *Client* – Remplacement d'*Evénements*	Utilise le *Service* HistoryUpdate pour remplacer des Evénements historiques.
Client	Accès à l'historique *Client* – Suppression d'*Evénements*	Utilise le *Service* HistoryUpdate pour supprimer des Evénements historiques.
Client	Accès à l'historique *Client* – Insertion de Données	Utilise le *Service* HistoryUpdate pour insérer des valeurs de données historiques.
Client	Accès à l'historique *Client* – Suppression de Données	Utilise le *Service* HistoryUpdate pour supprimer des valeurs de données historiques.
Client	Accès à l'historique *Client* – Mise à Jour de Données	Utilise le *Service* HistoryUpdate pour mettre à jour des valeurs de données historiques.
Client	Accès à l'historique *Client* – Remplacement de Données	Utilise le *Service* HistoryUpdate pour remplacer des valeurs de données historiques.
Client	Accès à l'historique *Client* – Annotations	Saisit et récupère des Annotations de données historiques. La récupération est effectuée à l'aide de la fonctionnalité de lecture de données historiques brutes normalisée (ReadRawModifiedDetails). La saisie utilise la fonctionnalité de mise à jour historique (UpdateStructureDataDetails) normalisée.
Client	Accès à l'historique *Client* – Instance Temporelle	Lit les données historiques à une instance spécifiée dans le temps à l'aide de la structure ReadAtTimeDetails.
Client	Accès à l'historique *Client* – Horodatage *Serveur*	Utilise le ServerTimestamp (ainsi que le SourceTimestamp par défaut), s'il est fourni par le *Serveur*.

Le Tableau 18 décrit les éléments relatifs aux Agrégats qui peuvent être intégrés dans les profils. Les *Serveurs* qui prennent en charge les Agrégats présentent cette fonctionnalité et les *Clients* qui utilisent les Agrégats mettent en œuvre une partie de la fonctionnalité.

Tableau 18 – Agrégats

Catégorie	Intitulé	Description
Serveur	Agrégat – Configuration maître	Prend en charge un *Objet* d'AggregateConfigurationType parmi les HistoricalServerCapabilities (définies dans l'IEC 62541-11).
Serveur	Agrégat – Configuration historique	Prend en charge au moins un *Objet* d'AggregateConfigurationType. Les *Objets* d'AggregateConfigurationType surviennent dans un *Objet* d'HistoricalConfiguration, permettant les configurations spécifiques aux *Variables*.
Serveur	Agrégat – Interpolative	Prend en charge l'Agrégat Interpolative pour l'Accès à l'historique.
Serveur	Agrégat – Average	Prend en charge l'Agrégat Average pour l'Accès à l'historique.
Serveur	Agrégat – TimeAverage	Prend en charge l'Agrégat TimeAverage pour l'Accès à l'historique.
Serveur	Agrégat – TimeAverage2	Prend en charge l'Agrégat TimeAverage2 pour l'Accès à l'historique.
Serveur	Agrégat – Total	Prend en charge l'Agrégat Total pour l'Accès à l'historique.
Serveur	Agrégat – Total2	Prend en charge l'Agrégat Total2 pour l'Accès à l'historique.
Serveur	Agrégat – Minimum	Prend en charge l'Agrégat Minimum pour l'Accès à l'historique
Serveur	Agrégat – MinimumActualTime	Prend en charge l'Agrégat MinimumActualTime pour l'Accès à l'historique.
Serveur	Agrégat – Minimum2	Prend en charge l'Agrégat Minimum2 pour l'Accès à l'historique.
Serveur	Agrégat – MinimumActualTime2	Prend en charge l'Agrégat MinimumActualTime2 pour l'Accès à l'historique.
Serveur	Agrégat – Maximum	Prend en charge l'Agrégat Maximum pour l'Accès à l'historique.
Serveur	Agrégat – MaximumActualTime	Prend en charge l'Agrégat MaximumActualTime pour l'Accès à l'historique.
Serveur	Agrégat – Maximum2	Prend en charge l'Agrégat Maximum2 pour l'Accès à l'historique.
Serveur	Agrégat – MaximumActualTime2	Prend en charge l'Agrégat MaximumActualTime2 pour l'Accès à l'historique.
Serveur	Agrégat – Range	Prend en charge l'Agrégat Range pour l'Accès à l'historique.
Serveur	Agrégat – Range2	Prend en charge l'Agrégat Range2 pour l'Accès à l'historique.
Serveur	Agrégat – Count	Prend en charge l'Agrégat Count pour l'Accès à l'historique.
Serveur	Agrégat – DurationInStateZero	Prend en charge l'Agrégat DurationInStateZero pour l'Accès à l'historique.
Serveur	Agrégat – DurationInStateNonZero	Prend en charge l'Agrégat DurationInStateNonZero pour l'Accès à l'historique.
Serveur	Agrégat – NumberOfTransitions	Prend en charge l'Agrégat NumberOfTransitions pour l'Accès à l'historique.
Serveur	Agrégat – Start	Prend en charge l'Agrégat Start pour l'Accès à l'historique.
Serveur	Agrégat – StartBound	Prend en charge l'Agrégat StartBound pour l'Accès à l'historique.
Serveur	Agrégat – End	Prend en charge l'Agrégat End pour l'Accès à l'historique.
Serveur	Agrégat – EndBound	Prend en charge l'Agrégat EndBound pour l'Accès à l'historique.
Serveur	Agrégat – Delta	Prend en charge l'Agrégat Delta pour l'Accès à l'historique.
Serveur	Agrégat – DeltaBounds	Prend en charge l'Agrégat DeltaBounds pour l'Accès à l'historique.
Serveur	Agrégat – DurationGood	Prend en charge l'Agrégat DurationGood pour l'Accès à l'historique.
Serveur	Agrégat – DurationBad	Prend en charge l'Agrégat DurationBad pour l'Accès à l'historique.
Serveur	Agrégat – PercentGood	Prend en charge l'Agrégat PercentGood pour l'Accès à l'historique.
Serveur	Agrégat – PercentBad	Prend en charge l'Agrégat PercentBad pour l'Accès à l'historique.
Serveur	Agrégat – WorstQuality	Prend en charge l'Agrégat WorstQuality pour l'Accès à l'historique.

Catégorie	Intitulé	Description
Serveur	Agrégat – WorstQuality2	Prend en charge l'Agrégat WorstQuality2 pour l'Accès à l'historique.
Serveur	Agrégat – AnnotationCount	Prend en charge l'Agrégat AnnotationCount pour l'Accès à l'historique.
Serveur	Agrégat – StandardDeviationSample	Prend en charge l'Agrégat StandardDeviationSample pour l'Accès à l'historique.
Serveur	Agrégat – VarianceSample	Prend en charge l'Agrégat VarianceSample pour l'Accès à l'historique.
Serveur	Agrégat – StandardDeviationPopulation	Prend en charge l'Agrégat StandardDeviationPopulation pour l'Accès à l'historique.
Serveur	Agrégat – VariancePopulation	Prend en charge l'Agrégat VariancePopulation pour l'Accès à l'historique.
Serveur	Agrégat – Custom	Le *Serveur* prend en charge les Agrégats personnalisés pour l'Accès à l'historique pour lesquels aucun essai normalisé n'est défini. Ces Agrégats sont indiqués comme non soumis à l'essai par cette *ConformanceUnit*.
Serveur	*Abonnement aux Agrégats* – Filter	Prend en charge les filtres pour l'abonnement "Agrégat", qui nécessite qu'au moins un des Agrégats définis soit pris en charge, comme défini dans l'IEC 62541-13.
Serveur	*Abonnement aux Agrégats* – Interpolative	Prend en charge le filtre d'abonnement pour l'Agrégat "Interpolative".
Serveur	*Abonnement aux Agrégats* – Average	Prend en charge le filtre d'abonnement pour l'Agrégat "Average".
Serveur	*Abonnement aux Agrégats* – TimeAverage	Prend en charge le filtre d'abonnement pour l'Agrégat "TimeAverage".
Serveur	*Abonnement aux Agrégats* – TimeAverage2	Prend en charge le filtre d'abonnement pour l'Agrégat "TimeAverage2".
Serveur	*Abonnement aux Agrégats* – Total	Prend en charge le filtre d'abonnement pour l'Agrégat "Total".
Serveur	*Abonnement* Agrégat – Total2	Prend en charge le filtre d'abonnement pour l'Agrégat "Total2".
Serveur	*Abonnement aux Agrégats* – Minimum	Prend en charge le filtre d'abonnement pour l'Agrégat "Minimum".
Serveur	*Abonnement aux Agrégats* – MinimumActualTime	Prend en charge le filtre d'abonnement pour l'Agrégat "MinimumActualTime".
Serveur	*Abonnement* Agrégat – Minimum2	Prend en charge le filtre d'abonnement pour l'Agrégat "Minimum2".
Serveur	*Abonnement* Agrégat – MinimumActualTime2	Prend en charge le filtre d'abonnement pour l'Agrégat "MinimumActualTime2".
Serveur	*Abonnement aux Agrégats* – Maximum	Prend en charge le filtre d'abonnement pour l'Agrégat "Maximum".
Serveur	*Abonnement aux Agrégats* – MaximumActualTime	Prend en charge le filtre d'abonnement pour l'Agrégat "MaximumActualTime".
Serveur	*Abonnement* Agrégat – Maximum2	Prend en charge le filtre d'abonnement pour l'Agrégat "Maximum2".
Serveur	*Abonnement* Agrégat – MaximumActualTime2	Prend en charge le filtre d'abonnement pour l'Agrégat "MaximumActualTime2".
Serveur	*Abonnement aux Agrégats* – Range	Prend en charge le filtre d'abonnement pour l'Agrégat "Range".
Serveur	*Abonnement* Agrégat – Range2	Prend en charge le filtre d'abonnement pour l'Agrégat "Range2".
Serveur	*Abonnement aux Agrégats* – Count	Prend en charge le filtre d'abonnement pour l'Agrégat "Count".
Serveur	*Abonnement aux Agrégats* – DurationInStateZero	Prend en charge le filtre d'abonnement pour l'Agrégat "DurationInStateZero".
Serveur	*Abonnement aux Agrégats* – DurationInStateNonZero	Prend en charge le filtre d'abonnement pour l'Agrégat "DurationInStateNonZero".

Catégorie	Intitulé	Description
Serveur	*Abonnement aux Agrégats* – NumberOfTransitions	Prend en charge le filtre d'abonnement pour l'Agrégat "NumberOfTransitions".
Serveur	*Abonnement aux Agrégats* – Start	Prend en charge le filtre d'abonnement pour l'Agrégat "Start".
Serveur	*Abonnement aux Agrégats* – StartBound	Prend en charge le filtre d'abonnement pour l'Agrégat "StartBound".
Serveur	*Abonnement aux Agrégats* – End	Prend en charge le filtre d'abonnement pour l'Agrégat "End".
Serveur	*Abonnement aux Agrégats* – EndBound	Prend en charge le filtre d'abonnement pour l'Agrégat "EndBound".
Serveur	*Abonnement aux Agrégats* – Delta	Prend en charge le filtre d'abonnement pour l'Agrégat "Delta".
Serveur	*Abonnement aux Agrégats* – DeltaBounds	Prend en charge le filtre d'abonnement pour l'Agrégat "DeltaBounds".
Serveur	*Abonnement aux Agrégats* – DurationGood	Prend en charge le filtre d'abonnement pour l'Agrégat "DurationGood".
Serveur	*Abonnement aux Agrégats* – DurationBad	Prend en charge le filtre d'abonnement pour l'Agrégat "DurationBad".
Serveur	*Abonnement aux Agrégats* – PercentGood	Prend en charge le filtre d'abonnement pour l'Agrégat "PercentGood".
Serveur	*Abonnement aux Agrégats* – PercentBad	Prend en charge le filtre d'abonnement pour l'Agrégat "PercentBad".
Serveur	*Abonnement aux Agrégats* – WorstQuality	Prend en charge le filtre d'abonnement pour l'Agrégat "WorstQuality".
Serveur	*Abonnement* Agrégat – WorstQuality2	Prend en charge le filtre d'abonnement pour l'Agrégat "WorstQuality2".
Serveur	*Abonnement aux Agrégats* – AnnotationCount	Prend en charge le filtre d'abonnement pour l'Agrégat "AnnotationCount".
Serveur	*Abonnement aux Agrégats* – StandardDeviationSample	Prend en charge le filtre d'abonnement pour l'Agrégat "StandardDeviationSample".
Serveur	*Abonnement aux Agrégats* – VarianceSample	Prend en charge le filtre d'abonnement pour l'Agrégat "VarianceSample".
Serveur	*Abonnement aux Agrégats* – StandardDeviationPopulation	Prend en charge le filtre d'abonnement pour l'Agrégat "StandardDeviationPopulation".
Serveur	*Abonnement aux Agrégats* – VariancePopulation	Prend en charge le filtre d'abonnement pour l'Agrégat "VariancePopulation".
Serveur	*Abonnement aux Agrégats* – Custom	Le *Serveur* prend en charge l'abonnement aux Agrégats personnalisés pour lesquels aucun essai normalisé n'est défini. Ces Agrégats sont indiqués comme non soumis à l'essai par cette *ConformanceUnit*.
Client	Agrégat – Usage *Client*	Utilise l'Accès à l'historique de l'Agrégat qui nécessite qu'au moins un des Agrégats définis soit pris en charge comme défini dans l'IEC 62541-13.
Client	Agrégat – Interpolative *Client*	Utilise l'Accès à l'historique de l'Agrégat Interpolative.
Client	Agrégat – Average *Client*	Utilise l'Accès à l'historique de l'Agrégat Average.
Client	Agrégat – TimeAverage *Client*	Utilise l'Accès à l'historique de l'Agrégat TimeAverage.
Client	Agrégat – TimeAverage2 *Client*	Utilise l'Accès à l'historique de l'Agrégat TimeAverage2.
Client	Agrégat – Total *Client*	Utilise l'Accès à l'historique de l'Agrégat Total.
Client	Agrégat – Total2 *Client*	Utilise l'Accès à l'historique de l'Agrégat Total2.
Client	Agrégat – Minimum *Client*	Utilise l'Accès à l'historique de l'Agrégat Minimum.
Client	Agrégat – MinimumActualTime *Client*	Utilise l'Accès à l'historique de l'Agrégat MinimumActualTime.
Client	Agrégat – Minimum2 *Client*	Utilise l'Accès à l'historique de l'Agrégat Minimum2.
Client	Agrégat – MinimumActualTime2 *Client*	Utilise l'Accès à l'historique de l'Agrégat MinimumActualTime2.

Catégorie	Intitulé	Description
Client	Agrégat – Maximum *Client*	Utilise l'Accès à l'historique de l'Agrégat Maximum.
Client	Agrégat – MaximumActualTime *Client*	Utilise l'Accès à l'historique de l'Agrégat MaximumActualTime.
Client	Agrégat – Maximum2 *Client*	Utilise l'Accès à l'historique de l'Agrégat Maximum2.
Client	Agrégat – MaximumActualTime2 *Client*	Utilise l'Accès à l'historique de l'Agrégat MaximumActualTime2.
Client	Agrégat – Range Client	Utilise l'Accès à l'historique de l'Agrégat Range.
Client	Agrégat – Range2 *Client*	Utilise l'Accès à l'historique de l'Agrégat Range2.
Client	Agrégat – Count *Client*	Utilise l'Accès à l'historique de l'Agrégat Count.
Client	Agrégat – DurationInStateZero *Client*	Utilise l'Accès à l'historique de l'Agrégat DurationInStateZero.
Client	Agrégat – DurationInStateNonZero *Client*	Utilise l'Accès à l'historique de l'Agrégat DurationInStateNonZero.
Client	Agrégat – NumberOfTransitions *Client*	Utilise l'Accès à l'historique de l'Agrégat NumberOfTransitions.
Client	Agrégat – Start *Client*	Utilise l'Accès à l'historique de l'Agrégat Start.
Client	Agrégat – StartBound *Client*	Utilise l'Accès à l'historique de l'Agrégat StartBound.
Client	Agrégat – End *Client*	Utilise l'Accès à l'historique de l'Agrégat End.
Client	Agrégat – EndBound *Client*	Utilise l'Accès à l'historique de l'Agrégat EndBound.
Client	Agrégat – Delta *Client*	Utilise l'Accès à l'historique de l'Agrégat Delta.
Client	Agrégat – DeltaBounds *Client*	Utilise l'Accès à l'historique de l'Agrégat DeltaBounds.
Client	Agrégat – DurationGood *Client*	Utilise l'Accès à l'historique de l'Agrégat DurationGood.
Client	Agrégat – DurationBad *Client*	Utilise l'Accès à l'historique de l'Agrégat DurationBad.
Client	Agrégat – PercentGood *Client*	Utilise l'Accès à l'historique de l'Agrégat PercentGood.
Client	Agrégat – PercentBad *Client*	Utilise l'Accès à l'historique de l'Agrégat PercentBad.
Client	Agrégat – WorstQuality *Client*	Utilise l'Accès à l'historique de l'Agrégat WorstQuality.
Client	Agrégat – WorstQuality2 *Client*	Utilise l'Accès à l'historique de l'Agrégat WorstQuality2.
Client	Agrégat – AnnotationCount *Client*	Utilise l'Accès à l'historique de l'Agrégat AnnotationCount.
Client	Agrégat – StandardDeviationSample *Client*	Utilise l'Accès à l'historique de l'Agrégat StandardDeviationSample.
Client	Agrégat – VarianceSample *Client*	Utilise l'Accès à l'historique de l'Agrégat VarianceSample.
Client	Agrégat – StandardDeviationPopulation *Client*	Utilise l'Accès à l'historique de l'Agrégat StandardDeviationPopulation.
Client	Agrégat – VariancePopulation *Client*	Utilise l'Accès à l'historique de l'Agrégat VariancePopulation.
Client	Agrégat – Agrégats personnalisés *Client*	Le *Client* peut utiliser tous les Agrégats personnalisés dans la liste d'Agrégats, via l'Accès à l'historique, présenté par le *Serveur*. Cela comprend l'affichage ou l'utilisation de données d'une certaine manière.
Client	*Abonnement aux Agrégats* – Filter *Client*	S'abonne à des données utilisant des filtres d'Agrégats qui nécessitent qu'au moins un des Agrégats définis dans l'IEC 62541-13 soit pris en charge.
Client	*Abonnement aux Agrégats* – Interpolative *Client*	S'abonne à des données utilisant le filtre d'Agrégat Interpolative.
Client	*Abonnement aux Agrégats* – Average *Client*	S'abonne à des données utilisant le filtre d'Agrégat Average.
Client	*Abonnement aux Agrégats* – TimeAverage *Client*	S'abonne à des données utilisant le filtre d'Agrégat TimeAverage.
Client	*Abonnement aux Agrégats* – TimeAverage2 *Client*	S'abonne à des données utilisant le filtre d'Agrégat TimeAverage2.

Catégorie	Intitulé	Description
Client	*Abonnement aux Agrégats* – Total *Client*	S'abonne à des données utilisant le filtre d'Agrégat Total.
Client	*Abonnement aux Agrégats* – Total2 *Client*	S'abonne à des données utilisant le filtre d'Agrégat Total2.
Client	*Abonnement aux Agrégats* – Minimum *Client*	S'abonne à des données utilisant le filtre d'Agrégat Minimum.
Client	*Abonnement aux Agrégats* – MinimumActualTime *Client*	S'abonne à des données utilisant le filtre d'Agrégat MinimumActualTime.
Client	*Abonnement aux Agrégats* – Minimum2 *Client*	S'abonne à des données utilisant le filtre d'Agrégat Minimum2.
Client	*Abonnement aux Agrégats* – MinimumActualTime2 *Client*	S'abonne à des données utilisant le filtre d'Agrégat MinimumActualTime2.
Client	*Abonnement aux Agrégats* – Maximum *Client*	S'abonne à des données utilisant le filtre d'Agrégat Maximum.
Client	*Abonnement* aux Agrégats – MaximumActualTime *Client*	S'abonne à des données utilisant le filtre d'Agrégat MaximumActualTime.
Client	*Abonnement aux Agrégats* – MaximumActualTime2 *Client*	S'abonne à des données utilisant le filtre d'Agrégat MaximumActualTime2.
Client	*Abonnement aux Agrégats* – Maximum2 *Client*	S'abonne à des données utilisant le filtre d'Agrégat Maximum2.
Client	*Abonnement* aux Agrégats – Range *Client*	S'abonne à des données utilisant le filtre d'Agrégat Range.
Client	*Abonnement aux Agrégats* – Range2 *Client*	S'abonne à des données utilisant le filtre d'Agrégat Range2.
Client	*Abonnement* aux Agrégats – Count *Client*	S'abonne à des données utilisant le filtre d'Agrégat Count.
Client	*Abonnement* aux Agrégats – DurationInStateZero *Client*	S'abonne à des données utilisant le filtre d'Agrégat DurationInStateZero.
Client	*Abonnement* aux Agrégats – DurationInStateNonZero *Client*	S'abonne à des données utilisant le filtre d'Agrégat DurationInStateNonZero.
Client	*Abonnement* aux Agrégats – NumberOfTransitions *Client*	S'abonne à des données utilisant le filtre d'Agrégat NumberOfTransitions.
Client	*Abonnement* aux Agrégats – Start *Client*	S'abonne à des données utilisant le filtre d'Agrégat Start.
Client	*Abonnement* aux Agrégats – StartBound *Client*	S'abonne à des données utilisant le filtre d'Agrégat StartBound.
Client	*Abonnement* aux Agrégats – End *Client*	S'abonne à des données utilisant le filtre d'Agrégat End.
Client	*Abonnement* aux Agrégats – EndBound *Client*	S'abonne à des données utilisant le filtre d'Agrégat EndBound.
Client	*Abonnement* aux Agrégats – Delta *Client*	S'abonne à des données utilisant le filtre d'Agrégat Delta.
Client	*Abonnement* aux Agrégats – DeltaBounds *Client*	S'abonne à des données utilisant le filtre d'Agrégat DeltaBounds.
Client	*Abonnement* aux Agrégats – DurationGood *Client*	S'abonne à des données utilisant le filtre d'Agrégat DurationGood.
Client	*Abonnement* aux Agrégats – DurationBad *Client*	S'abonne à des données utilisant le filtre d'Agrégat DurationBad.
Client	*Abonnement* aux Agrégats – PercentGood *Client*	S'abonne à des données utilisant le filtre d'Agrégat PercentGood.
Client	*Abonnement* aux Agrégats – PercentBad *Client*	S'abonne à des données utilisant le filtre d'Agrégat PercentBad.
Client	*Abonnement* aux Agrégats – WorstQuality *Client*	S'abonne à des données utilisant le filtre d'Agrégat WorstQuality.
Client	*Abonnement* aux Agrégats – WorstQuality2 *Client*	S'abonne à des données utilisant le filtre d'Agrégat WorstQuality2.

Catégorie	Intitulé	Description
Client	*Abonnement* aux Agrégats – AnnotationCount *Client*	S'abonne à des données utilisant le filtre d'Agrégat AnnotationCount.
Client	*Abonnement* aux Agrégats – StandardDeviationSample *Client*	S'abonne à des données utilisant le filtre d'Agrégat StandardDeviationSample.
Client	*Abonnement* aux Agrégats – VarianceSample *Client*	S'abonne à des données utilisant le filtre d'Agrégat VarianceSample.
Client	*Abonnement* aux Agrégats – StandardDeviationPopulation *Client*	S'abonne à des données utilisant le filtre d'Agrégat StandardDeviationPopulation.
Client	*Abonnement* aux Agrégats – VariancePopulation *Client*	S'abonne à des données utilisant le filtre d'Agrégat VariancePopulation.
Client	*Abonnement* aux Agrégats – Custom Aggregates *Client*	Le *Client* prend en charge l'abonnement à tous les Agrégats personnalisés dans la liste des Agrégats présentés par le *Serveur*. Cela comprend l'affichage ou l'utilisation de données d'une certaine manière.

Le Tableau 19 décrit les éléments relatifs à l'Audit qui peuvent être intégrés dans les profils. La plupart des *Serveurs* à forte capacité de ressource prennent en charge ces caractéristiques, alors que certains *Serveurs* possédant des capacités de ressources inférieures peuvent ne pas fournir cette fonctionnalité. Les *Clients* sensibilisés à la sécurité ou habitués à prendre en charge la journalisation des activités de sécurité prennent en charge ces caractéristiques.

Tableau 19 – Audit

Catégorie	Intitulé	Description
Serveur	Audit – Base	Prend en charge les AuditEvents. La liste des événements d'audit pris en charge doit être vérifiée lors des essais de certification et est indiquée dans les résultats des essais de certification. Les AuditEvents de base sont définis dans l'IEC 62541-3 et dans l'IEC 62541-5.
Client	Audit *Client* – ID d'audit	Le *Client* prend en charge la génération des identificateurs d'AuditEvents et les fournit aux *Serveurs*.
Client	Audit *Client* – Abonnement	Le *Client* prend en charge l'abonnement aux AuditEvents, ainsi que leur archivage/traitement sécurisés.

Le Tableau 20 décrit les éléments relatifs à la Redondance qui sont intégrés dans les profils. Les *Serveurs* qui prennent en charge la redondance prennent également en charge les *ConformanceUnits* appropriées selon le type de redondance correspondant. Les *Clients* capables de gérer la redondance prennent en charge les *ConformanceUnits* appropriées selon le type de redondance pris en charge.

Tableau 20 – Redondance

Catégorie	Intitulé	Description
Serveur	Redondance *Serveur*	Prend en charge la redondance fondée sur le *Serveur*.
Serveur	Redondance *Serveur* Transparente	Prend en charge la redondance *Serveur* transparente.
Client	Redondance *Client*	Le *Client* prend en charge la redondance *Client*. Les *Clients* qui prennent en charge la redondance peuvent basculer sur un autre *Client* (exige une certaine communication hors bande).
Client	Redondance *Client* – Commutateur	Les *Clients* qui prennent en charge cette *ConformanceUnit* surveillent le statut de redondance dans le cas des *Serveurs* à redondance non transparente et basculent sur le *Serveur* auxiliaire lorsqu'ils identifient un changement dans le statut du serveur.

Le Tableau 21 décrit les éléments d'un Serveur "Découverte" Global (GDS – Global *Discovery Server*). Les *Serveurs* qui agissent comme des GDS prendraient en charge ces *ConformanceUnits*.

Tableau 21 – Serveur "Découverte" Global

Catégorie	Intitulé	Description
Service d'annuaire global	GDS Répertoire des applications	Prend en charge l'*Objet* Répertoire avec toutes les Méthodes telles que RegisterApplication et QueryServers.
Service d'annuaire global	GDS Applications de Requêtes	Prend en charge la *Méthode* QueryApplications sur l'Objet Répertoire spécifiée dans l'IEC 62541-12.
Service d'annuaire global	GDS Connectivité LDS-ME	Le GDS peut être configuré pour utiliser des installations LDS-ME spécifiques pour répertorier les applications de manière semi-automatique pour tous les *Serveurs* d'un sous-réseau.
Service d'annuaire global	GDS Gestionnaire de *Certificat* – Modèle Pull	Cette Unité de conformité exige la prise en charge de la totalité du Modèle d'information et des *Services* de gestion des *Certificats*, y compris le Modèle Pull spécifié dans l'IEC 62541-12.
Service d'annuaire global	GDS Gestionnaire de *Certificat* – Modèle Push	Cette Unité de conformité exige l'utilisation de la totalité du Modèle d'information et des *Services* pour le Modèle Push de la gestion des *Certificats*, spécifié dans l'IEC 62541-12.
Service d'annuaire global	GDS *Service* Authentifiants clés – Modèle Pull	Cette Unité de conformité exige la prise en charge de la totalité du Modèle d'information et des *Services* de gestion des KeyCredential en mode Pull, spécifié dans l'IEC 62541-12.
Service d'annuaire global	GDS *Service* Authentifiants clés – Modèle Push	Cette Unité de conformité exige l'utilisation de la totalité du Modèle d'information et des *Services* de gestion des KeyCredential en mode Push, spécifié dans l'IEC 62541-12.
Service d'annuaire global	GDS *Serveur* de *Services* d'Autorisation	Cette Unité de conformité exige la prise en charge des objets AuthorizationServiceType spécifiés dans l'IEC 62541-12. Les Clients UA utilisent la Méthode RequestAccessToken sur ces Objets pour demander un Jeton d'accès auprès d'un Fournisseur d'identité.

5.5 Divers

Le Tableau 22 décrit différentes *ConformanceUnits*.

Tableau 22 – Divers

Catégorie	Intitulé	Description
Serveur	Documentation – *Profils* pris en charge	La documentation comprend une description des profils pris en charge par le produit. Cette description inclut le niveau des essais de Certification auxquels le produit a été soumis avec succès.
Serveur	Documentation – Langages multiples	La documentation est disponible dans plusieurs langages. Les résultats de cette unité de conformité comprennent la liste des langages pris en charge.
Serveur	Documentation – Guide Utilisateur	L'application comprend la documentation qui décrit la fonctionnalité disponible fournie par l'application. Pour les *Serveurs*, elle comprend un récapitulatif de l'ensemble des fonctionnalités fournies par le *Serveur*.
Serveur	Documentation – En ligne	La documentation fournie par l'application est disponible au format électronique dans le cadre de l'application. La documentation électronique peut être une page web, un document installé ou un CD/DVD, mais dans tous les cas, l'accès s'effectue depuis l'application ou depuis un lien installé avec l'application.
Serveur	Documentation – Installation	L'application comprend les instructions d'installation suffisantes pour installer facilement l'application. Cela comprend des descriptions de tout élément de configuration possible. Les instructions pour charger ou configurer les éléments relatifs à la sécurité tels que les *Certificats* d'instance d'application.
Serveur	Documentation – Guide de Recherche de pannes	L'application comprend la documentation qui décrit les problèmes types qu'un utilisateur peut rencontrer, ainsi que les actions qu'il peut effectuer pour résoudre ces problèmes. La documentation peut aussi décrire les astuces ou autres actions susceptibles d'aider l'utilisateur à diagnostiquer ou à résoudre un problème. Elle peut aussi décrire les outils ou autres éléments qui peuvent être utilisés pour diagnostiquer ou résoudre les problèmes. Le Guide de recherche de pannes en lui-même peut faire partie d'une autre documentation, mais il convient qu'il soit suffisamment complet pour apporter des informations utiles à un utilisateur novice.
Client	Documentation *Client* – Profils pris en charge	La documentation comprend une description des profils pris en charge par le produit. Cette description inclut tous les certificats de logiciel qui décrivent le niveau des essais de Certification auxquels le produit a été soumis avec succès.
Client	Documentation *Client* – Langages multiples	La documentation est disponible dans plusieurs langages. Les résultats de cette unité de conformité comprennent la liste des langages pris en charge.
Client	Documentation *Client* – Guide Utilisateur	L'application comprend la documentation qui décrit la fonctionnalité disponible fournie par l'application. Pour les applications clientes, cela comprend toutes les restrictions concernant les opérateurs ou les fonctionnalités générales que l'application cliente utilise.
Client	Documentation *Client* – En ligne	La documentation fournie par l'application est disponible au format électronique dans le cadre de l'application. La documentation électronique peut être une page web, un document installé ou un CD/DVD, mais dans tous les cas, l'accès s'effectue depuis l'application ou depuis un lien installé avec l'application.
Client	Documentation *Client* – Installation	L'application comprend les instructions d'installation suffisantes pour installer facilement l'application. Cela comprend des descriptions de tout élément de configuration possible. Les instructions pour charger ou configurer les éléments relatifs à la sécurité tels que les *Certificats* d'instance d'application.
Client	Documentation *Client* – Guide de Recherche de pannes	L'application comprend la documentation qui décrit les problèmes types qu'un utilisateur peut rencontrer, ainsi que les actions qu'il peut effectuer pour résoudre ces problèmes. La documentation peut aussi décrire les astuces ou autres actions susceptibles d'aider l'utilisateur à diagnostiquer ou à résoudre un problème. Elle peut aussi décrire les outils ou autres éléments qui peuvent être utilisés pour diagnostiquer ou résoudre les problèmes. Le Guide de recherche de pannes en lui-même peut faire partie d'une autre documentation, mais il convient qu'il soit suffisamment complet pour apporter des informations utiles à un utilisateur novice.
Sécurité	Meilleures pratiques – Temporisations	L'utilisateur est capable de configurer des temporisations raisonnables pour les Canaux Sécurisés, les sessions et les abonnements, afin de limiter les problèmes de Refus de *Service* et de consommation des ressources (voir l'IEC 62541-2 pour plus de détails).
Sécurité	Meilleures pratiques – Gestion stricte des messages	L'application garantit que les messages formés de façon illégale ou incorrecte sont rejetés avec des codes d'erreur appropriés ou des actions appropriées comme spécifié dans l'IEC 62541-4 et dans l'IEC 62541-6.

Catégorie	Intitulé	Description
Sécurité	Meilleures pratiques – Nombres aléatoires	Tous les nombres aléatoires exigés pour l'utilisation en toute sécurité de générateurs de nombres aléatoires fondés sur une bibliothèque cryptographique appropriée.
Sécurité	Meilleures pratiques – Accès administratif	Le *Serveur* et le *Client* permettent une restriction appropriée de l'accès du personnel administratif. Cela comprend plusieurs niveaux d'accès administratif aux plateformes qui prennent en charge plusieurs rôles d'administration (telles que Windows ou Linux).
Sécurité	Meilleures pratiques – Gestion d'*alarmes*	Il convient qu'un Serveur limite la fonctionnalité d'Alarme critique aux utilisateurs qui disposent des droits appropriés pour effectuer ces actions. Cela comprend la désactivation des alarmes, l'ordonnancement des alarmes et la génération de messages de dialogue. Cela comprend aussi d'autres fonctionnalités relatives à la sécurité, telles que le maintien de temporisations appropriées pour l'ordonnancement et les dialogues et le fait d'empêcher une surcharge de messages de dialogue.
Sécurité	Meilleures pratiques – Evénements d'audit	Les abonnements pour les événements d'audit sont limités au personnel autorisé. Un *Serveur* peut aussi rejeter un *Abonnement* pour les événements d'audit, qui n'est pas sur un éventuel Canal sécurisé disponible.
Sécurité	Meilleures pratiques – *Client* Evénements d'audit	Le système de suivi d'audit se connecte à un Serveur en utilisant un Canal sécurisé et avec les droits d'administration appropriés pour permettre un accès aux Evénements d'audit.

6 Profils

6.1 Vue d'ensemble

Le présent article fournit la liste des catégories au sein desquelles un *Profil* peut être regroupé, la liste des *Profils* nommés ainsi que la liste détaillée de chaque *Profil*, notamment les *ConformanceUnits* définies directement et les *sous-profils* inclus dans le *Profil*.

6.2 Liste des profils

Le Tableau 23 répertorie les *Profils*. Le tableau répertorie les profils par catégorie, puis dans l'ordre alphabétique (selon le nom de *Profil*). Le tableau comprend une liste des catégories auxquelles le *Profil* est associé, ainsi qu'un URI. Ce dernier permet d'identifier de manière unique un *Profil*. L'URI doit pouvoir être utilisé pour accéder aux informations fournies dans ce document selon le *Profil* donné dans un affichage en ligne.

Une application (*Client* ou *Serveur*) doit mettre en œuvre toutes les *ConformanceUnits* dans un *Profil* afin d'être conforme à ce dernier. Certains *Profils* comportent des *ConformanceUnits* facultatives. Une *ConformanceUnit* facultative signifie qu'une application a la possibilité de ne pas prendre en charge la *ConformanceUnit*. Toutefois, si elle est prise en charge, l'application doit satisfaire à tous les essais associés à la *ConformanceUnit*. Par exemple, certaines *ConformanceUnits* exigent la disponibilité d'éléments de modèle d'information spécifiques. Elles sont par conséquent définies comme facultatives, de manière à pouvoir ignorer les éléments du modèle d'information. Si un *Serveur* souhaite être classé comme prenant en charge la *ConformanceUnit* facultative, il doit alors inclure tous les éléments de modèle d'information exigés dans la configuration fournie pour les essais de certification. Le résultat de l'essai généré par l'essai de certification dresse la liste de toutes les *ConformanceUnits* facultatives et indique si elles sont prises en charge ou non par l'application UA à l'essai. Certaines *ConformanceUnits* comprennent également les listes des DataTypes pris en charge ou des Sous-types facultatifs pris en charge; les listes sont gérées de la même façon que les *ConformanceUnits* facultatives. Toutes les exigences de consignation pour les *ConformanceUnits* facultatives s'appliquent également à ces listes de DataTypes ou de sous-types pris en charge.

Tableau 23 – Liste des Profils

Profil	Catégorie associée	URI
Facette Serveur 2017 principal	Serveur	http://opcfoundation.org/UA-Profile/Server/Core2017Facet
Facette Serveur Sans session	*Serveur*	http://opcfoundation.org/UA-*Profile*/*Server*/SessionLess
Facette Serveur Connexion inversée	Serveur	http://opcfoundation.org/UA-Profile/Server/ReverseConnect
Facette Serveur Comportement de base du serveur	Serveur	http://opcfoundation.org/UA-Profile/Server/Behaviour
Facette Serveur Demande modification d'état	Serveur	http://opcfoundation.org/UA-Profile/Server/RequestStateChange
Facette Serveur Découverte sous-réseau	Serveur	http://opcfoundation.org/UA-Profile/Server/SubnetDiscovery
Facette Serveur Gestion globale des certificats	Serveur	http://opcfoundation.org/UA-Profile/Server/GlobalCertificateManagement
Facette Serveur Service d'autorisation	Serveur	http://opcfoundation.org/UA-Profile/Server/AuthorizationServiceConfiguration
Facette Serveur Service KeyCredential	Serveur	http://opcfoundation.org/UA-Profile/Server/KeyCredentialManagement
Facette Serveur Attribut WriteMask	Serveur	http://opcfoundation.org/UA-Profile/Server/AttributeWriteMask
Facette Serveur Accès fichier	Serveur	http://opcfoundation.org/UA-Profile/Server/FileAccess
Facette Serveur Documentation	Serveur	http://opcfoundation.org/UA-Profile/Server/Documentation
Facette Serveur Abonnement intégré aux DataChanges	Serveur	http://opcfoundation.org/UA-Profile/Server/EmbeddedDataChangeSubscription
Facette Serveur 2017 Abonnement normalisé aux DataChanges	Serveur	http://opcfoundation.org/UA-Profile/Server/StandardDataChangeSubscription2017
Facette Serveur 2017 Abonnement amélioré aux DataChanges	Serveur	http://opcfoundation.org/UA-Profile/Server/EnhancedDataChangeSubscription2017
Facette Serveur Abonnement durable	Serveur	http://opcfoundation.org/UA-Profile/Server/DurableSubscription
Facette Serveur Accès aux données	Serveur	http://opcfoundation.org/UA-Profile/Server/DataAccess
Facette Serveur 2017 ComplexType	Serveur	http://opcfoundation.org/UA-Profile/Server/ComplexTypes2017
Facette Serveur Abonnement normalisé aux Événements	Serveur	http://opcfoundation.org/UA-Profile/Server/StandardEventSubscription
Facette Serveur Notification de l'espace d'adressage	Serveur	http://opcfoundation.org/UA-Profile/Server/AddressSpaceNotifier
Facette Serveur A & C Condition de base	Serveur	http://opcfoundation.org/UA-Profile/Server/ACBaseCondition
Facette Serveur A & C Refresh2	Serveur	http://opcfoundation.org/UA-Profile/Server/ACRefresh2
Facette Serveur A & C Instance de l'espace d'adressage	Serveur	http://opcfoundation.org/UA-Profile/Server/ACAddressSpaceInstance
Facette Serveur A & C Activation	Serveur	http://opcfoundation.org/UA-Profile/Server/ACEnable
Facette Serveur A & C AlarmMetrics	Serveur	http://opcfoundation.org/UA-Profile/Server/ACAlarmMetrics
Facette Serveur A & C Alarme	Serveur	http://opcfoundation.org/UA-Profile/Server/ACAlarm
Facette Serveur A & C Alarme acquittable	Serveur	http://opcfoundation.org/UA-Profile/Server/ACAckAlarm
Facette Serveur A & C Alarme exclusive	Serveur	http://opcfoundation.org/UA-Profile/Server/ACExclusiveAlarming
Facette Serveur A & C Alarme non exclusive	Serveur	http://opcfoundation.org/UA-Profile/Server/ACNon-ExclusiveAlarming
Facette Serveur A & C Instances précédentes	Serveur	http://opcfoundation.org/UA-Profile/Server/ACPreviousInstances

Profil	Catégorie associée	URI
Facette Serveur A & C Dialogue	Serveur	http://opcfoundation.org/UA-Profile/Server/ACDialog
Facette Serveur A & C CertificateExpiration	Serveur	http://opcfoundation.org/UA-Profile/Server/ACCertificateExpiration
A & E Facette Conteneur	Serveur	http://opcfoundation.org/UA-Profile/Server/AEWrapper
Facette Serveur Méthode	Serveur	http://opcfoundation.org/UA-Profile/Server/Methods
Facette Serveur Audit	Serveur	http://opcfoundation.org/UA-Profile/Server/Auditing
Facette Serveur Gestion des nœuds	Serveur	http://opcfoundation.org/UA-Profile/Server/NodeManagement
Facette Serveur Rôle de base utilisateur	Serveur	http://opcfoundation.org/UA-Profile/Server/UserRoleBase
Facette Serveur Gestion des rôles d'utilisateurs	Serveur	http://opcfoundation.org/UA-Profile/Server/UserRoleManagement
Facette Serveur Diagramme d'état	Serveur	http://opcfoundation.org/UA-*Profile*/*Server*/StateMachine
Facette Serveur Redondance client	Serveur	http://opcfoundation.org/UA-*Profile*/*Server*/ClientRedundancy
Facette Serveur Redondance transparente	Serveur	http://opcfoundation.org/UA-*Profile*/*Server*/TransparentRedundancy
Facette Serveur Redondance visible	Serveur	http://opcfoundation.org/UA-*Profile*/*Server*/VisibleRedundancy
Facette Serveur Données brutes historiques	Serveur	http://opcfoundation.org/UA-*Profile*/*Server*/HistoricalRawData
Facette Serveur Agrégat historique	Serveur	http://opcfoundation.org/UA-*Profile*/*Server*/AggregateHistorical
Facette Serveur Données historiques à temps	Serveur	http://opcfoundation.org/UA-Profile/Server/HistoricalDataAtTime
Facette Serveur Accès à l'historique Données modifiées	Serveur	http://opcfoundation.org/UA-Profile/Server/HistoricalModifiedData
Facette Serveur Annotation Historique	Serveur	http://opcfoundation.org/UA-Profile/Server/HistoricalAnnotation
Facette Serveur Insertion Données historiques	Serveur	http://opcfoundation.org/UA-Profile/Server/HistoricalDataInsert
Facette Serveur Mise à Jour Données historiques	Serveur	http://opcfoundation.org/UA-Profile/Server/HistoricalDataUpdate
Facette Serveur Remplacement données historiques	Serveur	http://opcfoundation.org/UA-Profile/Server/HistoricalDataReplace
Facette Serveur Suppression données historiques	Serveur	http://opcfoundation.org/UA-Profile/Server/HistoricalDataDelete
Facette Serveur Accès à l'historique Données structurées	Serveur	http://opcfoundation.org/UA-Profile/Server/HistoricalStructuredData
Facette Serveur Evénement Historique de base	Serveur	http://opcfoundation.org/UA-Profile/Server/BaseHistoricalEvent
Facette Serveur Mise à jour Evénement historique	Serveur	http://opcfoundation.org/UA-Profile/Server/HistoricalEventUpdate
Facette Serveur Remplacement Evénement historique	Serveur	http://opcfoundation.org/UA-Profile/Server/HistoricalEventReplace
Facette Serveur Insertion Evénement historique	Serveur	http://opcfoundation.org/UA-Profile/Server/HistoricalEventInsert
Facette Serveur Suppression Evénement historique	Serveur	http://opcfoundation.org/UA-Profile/Server/HistoricalEventDelete
Facette Serveur Abonnement aux agrégats	Serveur	http://opcfoundation.org/UA-Profile/Server/AggregateSubscription
Profil Serveur 2017 Dispositif Nano-Intégré	Serveur	http://opcfoundation.org/UA-Profile/Server/NanoEmbeddedDevice2017

Profil	Catégorie associée	URI
Profil Serveur 2017 Dispositif Micro-Intégré	Serveur	http://opcfoundation.org/UA-Profile/Server/MicroEmbeddedDevice2017
Profil Serveur UA 2017 intégré	Serveur	http://opcfoundation.org/UA-Profile/Server/EmbeddedUA2017
Profil Serveur UA 2017 normalisé	Serveur	http://opcfoundation.org/UA-Profile/Server/StandardUA2017
Facette *Client* 2017 principal	*Client*	http://opcfoundation.org/UA-*Profile*/*Client*/Core2017
Facette *Client* Sans Session	*Client*	http://opcfoundation.org/UA-*Profile*/*Client*/SessionLess
Facette *Client* Connexion inversée	Client	http://opcfoundation.org/UA-Profile/Client/ReverseConnect
Facette Client Comportement de base	Client	http://opcfoundation.org/UA-Profile/Client/Behaviour
Facette Client Découverte	Client	http://opcfoundation.org/UA-Profile/Client/Discovery
Facette Client Découverte Sous Réseau	Client	http://opcfoundation.org/UA-Profile/Client/SubnetDiscovery
Facette Client Découverte Globale	Client	http://opcfoundation.org/UA-Profile/Client/GlobalDiscovery
Facette Client Gestion Globale des Certificats	Client	http://opcfoundation.org/UA-Profile/Client/GlobalCertificateManagement
Facette Client Service KeyCredential	Client	http://opcfoundation.org/UA-Profile/Client/KeyCredentialManagement
Facette Client Demande de jeton d'accès	Client	http://opcfoundation.org/UA-Profile/Client/AccessTokenRequest
Facette Client Consultation de l'AddressSpace	Client	http://opcfoundation.org/UA-Profile/Client/AddressSpaceLookup
Facette Client Demande Modification d'Etat	Client	http://opcfoundation.org/UA-Profile/Client/RequestStateChange
Facette Client Accès Fichier	Client	http://opcfoundation.org/UA-Profile/Client/FileAccess
Facette Client 2015 Prise en Charge Entrée de Gamme	Client	http://opcfoundation.org/UA-Profile/Client/EntryLevelSupport2015
Facette Client Connexion Multi-Serveur	Client	http://opcfoundation.org/UA-Profile/Client/MultiServer
Documentation – Client	Client	http://opcfoundation.org/UA-Profile/Client/Documentation
Facette Client Attribut Lecture	Client	http://opcfoundation.org/UA-Profile/Client/AttributeRead
Facette Client Attribut Ecriture	Client	http://opcfoundation.org/UA-Profile/Client/AttributeWrite
Facette Client Abonné aux modifications de données	Client	http://opcfoundation.org/UA-Profile/Client/DataChangeSubscriber
Facette Client Abonnement durable	Client	http://opcfoundation.org/UA-Profile/Client/DurableSubscription
Facette Client Accès aux données	Client	http://opcfoundation.org/UA-Profile/Client/DataAccess
Facette Client Abonné aux Evénements	Client	http://opcfoundation.org/UA-Profile/Client/EventSubscriber
Facette Client Traitement des événements de base	Client	http://opcfoundation.org/UA-Profile/Client/BaseEventProcessing
Facette Client Hiérarchie de Notification et de Source	Client	http://opcfoundation.org/UA-Profile/Client/NotifierAndSourceHierarchy
Facette Client A & C Condition de base	Client	http://opcfoundation.org/UA-Profile/Client/ACBaseCondition
Facette Client A & C Refresh2	Client	http://opcfoundation.org/UA-Profile/Client/ACRefresh2
Facette Client A & C Instance de l'Espace d'adressage	Client	http://opcfoundation.org/UA-Profile/Client/ACAddressSpaceInstance
Facette Client A & C Activer	Client	http://opcfoundation.org/UA-Profile/Client/ACEnable
Facette Client A & C AlarmMetrics	Client	http://opcfoundation.org/UA-Profile/Client/ACAlarmMetrics
Facette Client A & C Alarme	Client	http://opcfoundation.org/UA-Profile/Client/ACAlarm

Profil	Catégorie associée	URI
Facette Client A & C Alarme exclusive	Client	http://opcfoundation.org/UA-Profile/Client/ACExclusiveAlarming
Facette Client A & C Alarme non exclusive	Client	http://opcfoundation.org/UA-Profile/Client/ACNon-ExclusiveAlarming
Facette Client A & C Instances précédentes	Client	http://opcfoundation.org/UA-Profile/Client/ACPreviousInstances
Facette Client A & C Dialogue	Client	http://opcfoundation.org/UA-Profile/Client/ACDialog
Facette Client A & C CertificateExpiration	Client	http://opcfoundation.org/UA-Profile/Client/ACCertificateExpiration
A & E Facette Proxy	Client	http://opcfoundation.org/UA-Profile/Client/AEProxy
Facette Client Méthode	Client	http://opcfoundation.org/UA-Profile/Client/Method
Facette Client Audit	Client	http://opcfoundation.org/UA-Profile/Client/Auditing
Facette Client Gestion des Nœuds	Client	http://opcfoundation.org/UA-Profile/Client/NodeManagement
Facette Client Programmation de type avancé	Client	http://opcfoundation.org/UA-Profile/Client/TypeProgramming
Facette Client Gestion des Rôles d'Utilisateurs	Client	http://opcfoundation.org/UA-Profile/Client/UserRoleManagement
Facette Client Diagramme d'état	Client	http://opcfoundation.org/UA-Profile/Client/StateMachine
Facette Client Diagnostic	Client	http://opcfoundation.org/UA-Profile/Client/Diagnostic
Facette Client Redondant	Client	http://opcfoundation.org/UA-Profile/Client/Redundancy
Facette Client Commutateur de redondance	Client	http://opcfoundation.org/UA-Profile/Client/RedundancySwitch
Facette Client Accès à l'historique	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalAccess
Facette Client Données Historiques A Temps	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalAccessAtTime
Facette Client Agrégat Historique	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalAccessAggregate
Facette Client Annotation Historique	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalAnnotation
Facette Client Accès à l'historique Données Modifiées	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalAccessModifiedData
Facette Client Insérer Données Historiques	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalInsertData
Facette Client Mettre à Jour Données Historiques	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalUpdateData
Facette Client Remplacement données historiques	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalReplaceData
Facette Client Supprimer données structurées historiques	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalDeleteData
Facette Client Accès à l'historique Horodatage Serveur	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalServerTimeStamp
Facette Client Accès données structurées historiques	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalAccessStructuredData
Facette Client Données structurées historiques AtTime	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalAtTimeStructuredData
Facette Client Données structurées historiques modifiées	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalModifiedStructuredData
Facette Client Insérer données structurées historiques	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalInsertStructuredData
Facette Client Mettre à Jour données structurées historiques	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalUpdateStructuredData

Profil	Catégorie associée	URI
Facette Client Remplacer données structurées historiques	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalReplaceStructuredData
Facette Client Supprimer données structurées historiques	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalDeleteStructuredData
Facette Client Evénements Historiques	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalEvents
Facette Client Insertion Evénements Historiques	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalInsertEvents
Facette Client Mise à Jour Evénement Historique	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalUpdateEvents
Facette Client Remplacement Evénements Historiques	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalReplaceEvents
Facette Client Suppression Evénements Historiques	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalDeleteEvents
Facette Client Abonnement aux Agrégats	Client	http://opcfoundation.org/UA-Profile/Client/AggregateSubscriber
profil client UA normalisé;	Client	http://opcfoundation.org/UA-Profile/Client/Standard
Profil 2017 Client UA normalisé	Client	http://opcfoundation.org/UA-Profile/Client/Standard2017
Profil UA binaire UA-TCP UA-SC	Transport	http://opcfoundation.org/UA-Profile/Transport/uatcp-uasc-uabinary
Profil UA binaire HTTPS	Transport	http://opcfoundation.org/UA-Profile/Transport/https-uabinary
Protocole de transport XML UA HTTPS	Transport	http://opcfoundation.org/UA-Profile/Transport/https-uasoapxml
Protocole de transport UA JSON HTTPS	Transport	http://opcfoundation.org/UA-Profile/Transport/https-uajson
Profil UA binaire WSS UA-SC	Transport	http://opcfoundation.org/UA-Profile/Transport/wss-uasc-uabinary
Protocole de transport UA JSON WSS	Transport	http://opcfoundation.org/UA-Profile/Transport/wss-uajson
Sécurité utilisateur Contrôle accès complet	Sécurité, Serveur	http://opcfoundation.org/UA-Profile/Security/UserAccessFull
Sécurité utilisateur Contrôle accès de base	Sécurité, Serveur	http://opcfoundation.org/UA-Profile/Security/UserAccessBase
Synchronisation Temporelle Sécurité	Sécurité	http://opcfoundation.org/UA-Profile/Security/TimeSync
Meilleures pratiques – Evénements d'audit	Sécurité, Serveur	http://opcfoundation.org/UA-Profile/Security/BestPracticeAuditEvents
Meilleures pratiques – Gestion d'alarmes	Sécurité, Serveur	http://opcfoundation.org/UA-Profile/Security/BestPracticeAlarmHandling
Meilleures pratiques – Nombres aléatoires	Sécurité	http://opcfoundation.org/UA-Profile/Security/BestPracticeRandomNumbers
Meilleures pratiques – Temporisations	Sécurité	http://opcfoundation.org/UA-Profile/Security/BestPracticeTimeouts
Meilleures pratiques – Accès administratif	Sécurité	http://opcfoundation.org/UA-Profile/Security/BestPracticeAdministrativeAccess
Meilleures pratiques – Gestion stricte des messages	Sécurité, Serveur	http://opcfoundation.org/UA-Profile/Security/BestPracticeStrictMessage
Meilleures pratiques – Client Evénements d'audit	Client, Sécurité	http://opcfoundation.org/UA-Profile/Security/BestPracticeAuditEventsClient
TransportSecurity – TLS 1.2	Sécurité	http://opcfoundation.org/ UA-Profile/TransportSecurity/TLS-1-2
TransportSecurity – TLS 1.2 avec PFS	Sécurité	http://opcfoundation.org/UA-Profile/TransportSecurity/TLS-1-2-PFS
SecurityPolicy – None	Sécurité	http://opcfoundation.org/UA/SecurityPolicy#None

Profil	Catégorie associée	URI
SecurityPolicy [A] – Aes128-Sha256-RsaOaep	Sécurité	http://opcfoundation.org/UA/SecurityPolicy#Aes128_Sha256_RsaOaep
SecurityPolicy [B] – Basic256Sha256)	Sécurité	http://opcfoundation.org/UA/SecurityPolicy#Basic256Sha256
SecurityPolicy – Aes256-Sha256-RsaPss	Sécurité	http://opcfoundation.org/UA/SecurityPolicy#Aes256_Sha256_RsaPss
Facette Jeton Utilisateur – Anonyme	Sécurité	http://opcfoundation.org/UA-Profile/Security/UserToken/Anonymous
Facette Serveur Jeton utilisateur – Nom d'utilisateur Mot de passe	Sécurité, Serveur	http://opcfoundation.org/UA-Profile/Security/UserToken/Server/UserNamePassword
Facette Serveur Jeton utilisateur – Certificat X509	Sécurité, Serveur	http://opcfoundation.org/UA-Profile/Security/UserToken/Server/X509Certificate
Facette Serveur Jeton utilisateur – Jeton émis	Sécurité, Serveur	http://opcfoundation.org/UA-Profile/Security/UserToken/Server/IssuedToken
Facette Serveur Windows Jeton utilisateur – Jeton émis	Sécurité, Serveur	http://opcfoundation.org/UA-Profile/Security/UserToken/Server/IssuedTokenWindows
Facette Serveur Jeton utilisateur – JWT	Sécurité	http://opcfoundation.org/UA-Profile/Security/UserToken/Server/JsonWebToken
Facette Client Jeton Utilisateur – Nom d'Utilisateur Mot de Passe	Client, Sécurité	http://opcfoundation.org/UA-Profile/Security/UserToken/Client/UserNamePassword
Facette Client Jeton Utilisateur – Certificat X509	Client, Sécurité	http://opcfoundation.org/UA-Profile/Security/UserToken/Client/X509Certificate
Facette Client Jeton Utilisateur – Jeton Emis	Client, Sécurité	http://opcfoundation.org/UA-Profile/Security/UserToken/Client/IssuedToken
Facette Client Windows Jeton Utilisateur —f Jeton Emis	Client, Sécurité	http://opcfoundation.org/UA-Profile/Security/UserToken/Client/IssuedTokenWindows
Facette Client Jeton Utilisateur – JWT	Sécurité	http://opcfoundation.org/UA-Profile/Security/UserToken/Client/JsonWebToken
Profil 2017 Serveur Découverte Globale	Service d'annuaire global, Serveur	http://opcfoundation.org/UA-Profile/Server/GlobalDiscovery2017
Serveur 2017 Découverte Globale et Gestion des Certificats	Service d'annuaire global, Serveur	http://opcfoundation.org/UA-Profile/Server/GlobalDiscoveryAndCertificateManagement2017
Profil 2017 Client Gestion Globale des Certificats	Client, Service d'annuaire global	http://opcfoundation.org/UA-Profile/Client/GlobalCertificateManagement2017
Facette Serveur Demande d'autorisation Service global	Service d'annuaire global	http://opcfoundation.org/UA-Profile/Server/GlobalServiceAuthorization
Facette Pull KeyCredential Service Global	Service d'annuaire global	http://opcfoundation.org/UA-Profile/Server/GlobalServiceKeyCredentials
Facette Push KeyCredential Service Global	Service d'annuaire global	http://opcfoundation.org/UA-Profile/Client/GlobalServiceKeyCredentials

Le contenu de chacun des *Profils* répertoriés est décrit en 6.6 et lorsque l'intitulé équivaut au nom du *Profil*. Chaque tableau peut contenir des références aux *Profils* et/ou *ConformanceUnits* supplémentaires. Le référencement à un *Profil* signifie son inclusion pleine et entière. Les *ConformanceUnits* sont référencées à l'aide de leur nom et de leur groupe de conformité. Pour les détails des *ConformanceUnits*, il convient que le lecteur examine les informations relatives aux *ConformanceUnits* à l'Article 5.

6.3 Conventions applicables aux définitions des profils

Les conventions de nommage suivantes sont associées aux *Profils*:

- les *Profils* destinés aux *Serveurs* OPC UA comportent le terme *Serveur* dans leurs titres;
- les *Profils* destinés aux *Clients* OPC UA comportent le terme *Client* dans leurs titres;
- le terme "Facette" dans le titre d'un *Profil* indique que le *Profil* observé est supposé faire partie intégrante d'un autre *Profil* plus important ou concerne un aspect spécifique d'OPC UA. Les *Profils* dont le titre comporte le terme "Facette" sont supposés être combinés à d'autres *Profils* afin de définir la fonctionnalité complète d'un *Serveur* ou d'un *Client* OPC UA.

6.4 Versionnage des Profils

Le versionnage du *Profil* est réalisé au moyen d'une convention de nommage. Lorsqu'un profil est révisé, l'année de la nouvelle révision est ajoutée au nom. Exemple:

Version 1	Facette *Serveur* principal
Version 2	Facette *Serveur* 2017 principal

6.5 Applications

Un fournisseur qui développe une application UA, qu'il s'agisse d'une application *Serveur* ou d'une application *Client*, doit examiner la liste des *Profils* disponibles. Le fournisseur doit, à partir de cette liste, choisir les *Profils* qui incluent la fonctionnalité exigée par l'application. Généralement, il s'agit de *Profils* multiples. La conformité à un *Profil* unique peut ne pas produire une application complète. Dans la plupart des cas, des *Profils* multiples sont nécessaires pour produire une application utilisable. Tous les *Serveurs* et *Clients* doivent prendre en charge au moins un *Profil* principal (*Facette serveur* principal ou *Facette client* principal) et au moins un *Profil* de transport.

Par exemple, une application *Client* IHM peut choisir de prendre en charge la "*Facette Client* principal", le *Profil* "UA binaire UA-TCP UA-SC", la "*Facette Client* Accès aux Données", la "*Facette Client* Abonné aux Modifications de Données" et la "*Facette Client Attribut* d'Ecriture". Si le *Client* doit être soumis à un essai par un *TestLab*, il prend alors également en charge le *Profil* "Comportement *Client* de base". Cette liste des *Profils* permet au *Client* de communiquer avec un *Serveur* OPC UA en utilisant UA-TCP/UA Sécurité/UA binaire. Il est ainsi possible de réaliser les opérations d'abonnement aux données, d'écriture de données et de prendre en charge le modèle de données DA. Il est également possible de suivre la meilleure pratique comme ligne directrice pour le comportement. La Figure 2 représente la hiérarchie de *Profils* que cette application peut contenir. Cette figure est fournie à titre informatif seulement (les *Profils* représentés peuvent varier).

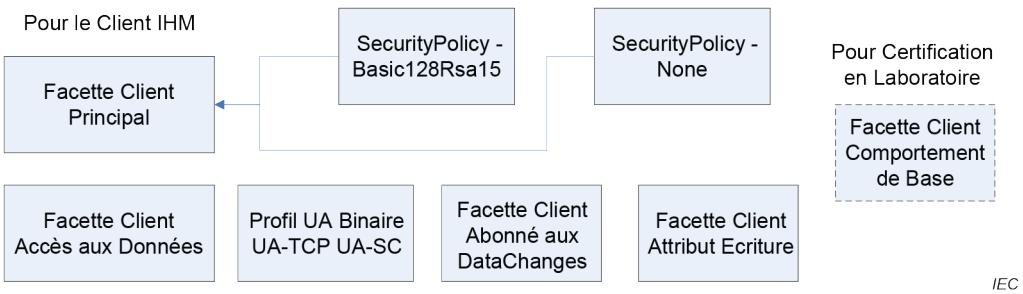


Figure 2 – Echantillon Client IHM

Il convient que tous les *Clients* prennent en compte les types de *Serveurs* et les *Profils de serveurs* qu'ils sont destinés à prendre en charge. Certains *Serveurs* peuvent ne pas prendre en charge les *Abonnements* et il convient que les *Clients* soient capables de se replier sur les *Services* de Lecture.

Un *Client* générique conçu pour communiquer avec plusieurs *Serveurs* et donc capable de remplir de nombreuses fonctionnalités est un cas spécial. Le "*Profil Client* UA normalisé" a été défini pour ce type de *Client*.

Toutefois, de nombreux *Clients* sont spécialisés et ne nécessitent pas l'ensemble des fonctionnalités du "*Profil Client* UA normalisé", mais prennent en charge seulement l'ensemble limité de fonctionnalités exigées. Un *Client* de tendance, par exemple, ne nécessiterait que des fonctionnalités permettant de s'abonner ou de lire les données.

Un autre exemple consiste en une application *Serveur* OPC UA d'un dispositif intégré pouvant choisir de prendre en charge le *Profil* "*Serveur* UA intégré" et le *Profil* "Facette *serveur* Accès aux données". Ce dispositif serait un dispositif à ressources limitées qui prendrait en charge les profils UA-TCP, UA-Sécurité, codage UA binaire, les abonnements aux données et le modèle de données DA. Il peut ne pas prendre en charge l'attribut facultatif d'écriture. La Figure 3 représente la hiérarchie que cette application peut contenir. Cette figure est fournie à titre informatif seulement (les *Profils* représentés peuvent varier).

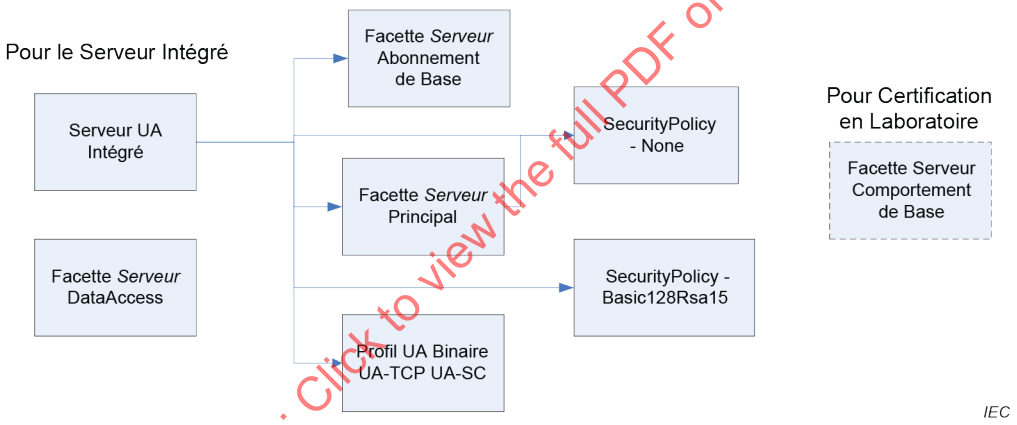


Figure 3 – Échantillon Serveur intégré

Une autre application *Serveur* système simple peut choisir de prendre en charge: le *Profil* "Serveur UA normalisé" et le *Profil* "Facette *serveur* DataAccess". Si le *Serveur* doit être soumis à un essai en laboratoire, il prend alors également en charge le *Profil* "Comportement de base du *Serveur*" ("Base *Server* Behaviour"). Ce dispositif serait un *Serveur* OPC UA de niveau moyen qui prendrait en charge tous les éléments pris en charge dans l'exemple précédent par le *Serveur* intégré. Il prendrait également en charge l'amélioration du service d'abonnement, ainsi que les opérations d'écriture. La Figure 4 représente la hiérarchie que cette application peut contenir. Cette figure est fournie à titre informatif seulement (les *Profils* représentés peuvent varier).

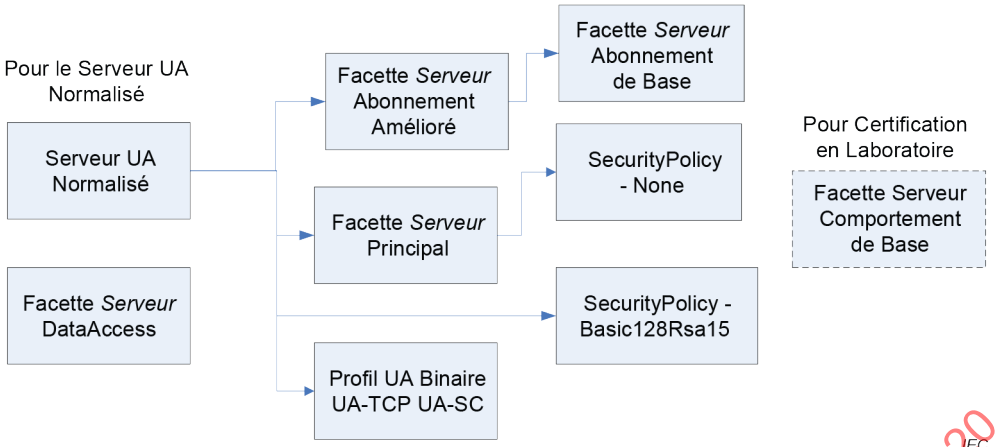


Figure 4 – Echantillon Serveur UA normalisé

Si l'interface IHM *Client* présentée en exemple était reliée à l'un ou l'autre des *Serveurs* présentés en exemple, elle peut devoir adapter son comportement sur la base du *Profil* signalé par les *Serveurs* respectifs. Si l'interface IHM *Client* communiquait avec le dispositif intégré, elle ne serait pas capable d'effectuer les opérations d'écriture. Il se peut également qu'elle doive limiter le nombre d'abonnements ou de sessions sur la base des limites de performance du *Serveur*. Si l'interface IHM *Client* était reliée au *Serveur* normalisé, elle serait capable d'ouvrir des fenêtres supplémentaires, présenterait de plus grandes limites concernant les éléments relatifs à la performance et pourrait autoriser les opérations d'écriture.

6.6 Tableaux de Profils

6.6.1 Généralités

Les 6.6.2 à 6.6.186 décrivent les *Profils* sous forme tabulaire.

Chaque tableau comporte trois colonnes. La première colonne est une description du groupe de conformité dont la *ConformanceUnit* fait partie intégrante. Ceci permet au lecteur de déterminer facilement la *ConformanceUnit*. Cette colonne peut également indiquer le "*Profil*" dans le cas, où l'élément indiqué n'est pas une *ConformanceUnit*, mais un *Profil* inclus. La deuxième colonne donne une description succincte de la *ConformanceUnit* ou du *Profil* inclus. La dernière colonne indique le caractère facultatif ou obligatoire de la *ConformanceUnit*.

6.6.2 Facette Serveur principal

La Facette Serveur principal n'est plus conseillée, car elle est annulée et remplacée par la Facette Serveur 2017 principal dans la v1.04.

6.6.3 Facette Serveur 2017 principal

Le Tableau 24 décrit les détails de la Facette *Serveur* 2017 principal. Cette Facette définit la fonctionnalité principale exigée pour toute mise en œuvre d'un *Serveur* UA. La fonctionnalité principale comporte la capacité de découvrir des points d'extrémité, établir des canaux de communication sécurisés, créer des Sessions, explorer l'*AddressSpace* et affecter des valeurs de lecture et/ou écriture aux *Attributs de Nœuds*. Les exigences essentielles sont les suivantes: prise en charge d'une *Session* unique, et prise en charge du *Serveur* et de l'*Objet* Capacités de ce dernier, ainsi que de tous les *Attributs* obligatoires pour les *Nœuds* dans l'*AddressSpace*, et l'authentification avec le UserName et le Mot de passe. Pour une application générale, il est recommandé que les *Serveurs* prennent en charge plusieurs *Profils* de transport et de sécurité. Cette Facette annule et remplace la "Facette *Serveur* principal".

Tableau 24 – Facette Serveur 2017 principal

Groupe	Titre Unité de conformité / *Profil*	Facultative
Profil	SecurityPolicy – None	False
Profil	Facette *Serveur* Jeton utilisateur – Nom d'utilisateur Mot de passe	False
Modèle d'espace d'adressage	Espace d'adressage – Atomicité	False
Modèle d'espace d'adressage	Espace d'adressage – Base	False
Modèle d'espace d'adressage	Espace d'adressage – Matrice complète seulement	False
Services Attribut	*Attribut* Lecture	False
Services Attribut	*Attribut* Ecriture d'Indices	True
Services Attribut	*Attribut* Ecriture des Valeurs	True
Informations de base	Informations de base – Structure principale	False
Informations de base	Informations de base – Temps de retour estimé	True
Informations de base	Informations de base – OptionSet	True
Informations de base	Informations de base – Règles de modélisation – Paramètre fictif	True
Informations de base	Informations de base – Liste de sélection	True
Informations de base	Informations de base – Capacités *Serveur*	True
Informations de base	Informations de base – ValueAsText	True
Services Découverte	*Découverte* "Trouver des *Serveurs* individuellement"	False
Services Découverte	*Découverte* "Obtenir des Points d'extrémité"	False
Sécurité	Sécurité Administration	True
Sécurité	Sécurité – Autorisation de Rôles *Serveur*	True
Services Session	*Session* "Base"	False
Services Session	*Session* "Comportement *Service* général"	False
Services Session	*Session* "Minimum 1"	False
Services Vue	Vue de base	False
Services Vue	Vue "Point de continuation minimum 01"	False
Services Vue	Vue "RegisterNodes"	False
Services Vue	Vue "TranslateBrowsePath"	False

6.6.4 Facette Serveur Sans session

Le Tableau 25 décrit les détails de la Facette *Serveur* Sans session. Définit l'utilisation de l'invocation du *Service* Sans Session dans un *Serveur*.

Tableau 25 – Facette Serveur Sans session

Groupe	Titre Unité de conformité / *Profil*	Facultative
Services Découverte	*Découverte* "Obtenir les points d'extrémité SessionLess"	False
Services Session	*Session* "Invocation Sans Session"	False

6.6.5 Facette Serveur Connexion inversée

Le Tableau 26 décrit les détails de la Facette *Serveur* Connexion Inversée. Cette Facette définit la prise en charge de la connectivité inversée dans un *Serveur*. Habituellement, une connexion est ouverte par le *Client* avant de démarrer le protocole d'établissement de liaison spécifique UA. Elle échoue, toutefois, lorsque les *Serveurs* sont protégés par des pare-feux sans accès ouvert auquel se connecter. Dans le scénario de connectivité inversée, le *Serveur* ouvre la connexion et commence avec un message ReverseHello demandant au *Client* d'établir un Canal sécurisé au moyen de cette connexion.

Tableau 26 – Facette Serveur Connexion inversée

Groupe	Titre Unité de conformité / *Profil*	Facultative
Protocole et Codage	Protocole de Connexion inversée *Serveur*	False

6.6.6 Facette Serveur Comportement de base du serveur

Le Tableau 27 décrit les détails de la Facette Comportement de base du *Serveur*. Cette Facette définit les meilleures pratiques en matière de configuration et de gestion des *Serveurs* déployés dans un environnement de production. Elle offre la capacité d'activer ou de désactiver certains protocoles et de configurer le *Serveur* "*Découverte*", ainsi que de spécifier le support sur lequel ce *Serveur* doit être enregistré.

Tableau 27 – Facette Serveur Comportement de base du serveur

Groupe	Titre Unité de conformité / *Profil*	Facultative
Services Découverte	*Découverte* "Configuration"	False
Protocole et Codage	Protocole Configuration	False
Sécurité	Sécurité Administration	False
Sécurité	Sécurité Administration – Schéma XML	False
Sécurité	Sécurité – Administration de Certification	False

6.6.7 Facette Serveur Demande modification d'état

Le Tableau 28 décrit les détails de la Facette *Serveur* Demande modification d'état. Cette Facette spécifie la prise en charge de la *Méthode* RequestServerStateChange.

Tableau 28 – Facette Serveur Demande modification d'état

Groupe	Titre Unité de conformité / *Profil*	Facultative
Informations de base	Informations de base – Méthode RequestServerStateChange	False

6.6.8 Facette Serveur Découverte sous-réseau

Le Tableau 29 décrit les détails de la Facette *Serveur Découverte* sous-Réseau. La prise en charge de cette Facette permet de découvrir le *Serveur* dans un sous-réseau au moyen du mDNS. Cette fonctionnalité est applicable uniquement si les *Serveurs* ne s'enregistrent pas avec un LDS.

Tableau 29 – Facette Serveur Découverte sous-Réseau

Groupe	Titre Unité de conformité / *Profil*	Facultative
Services Découverte	Découverte – Notification Serveur au moyen du mDNS	False

6.6.9 Facette Serveur Gestion globale des certificats

Le Tableau 30 décrit les détails de la Facette *Serveur* Gestion globale des *certificats*. Cette Facette définit la capacité à interagir avec un *Serveur* de Gestion Globale des *Certificats* pour obtenir un *Certificat* initial et des Listes de Confiance ou les renouveler.

Tableau 30 – Facette Serveur Gestion globale des certificats

Groupe	Titre Unité de conformité / *Profil*	Facultative
Sécurité	Modèle Push pour la Gestion Globale des *Certificats* et des TrustLists	False

6.6.10 Facette Serveur Service d'autorisation

Le Tableau 31 décrit les détails de la Facette *Serveur Service* d'autorisation. Cette Facette définit la prise en charge de la configuration des informations nécessaires pour valider les jetons d'accès lorsqu'ils sont présentés par un *Client* pendant l'établissement de la session. Les Jetons d'accès sont émis par les *Services* d'autorisation.

Tableau 31 – Facette Serveur Service d'autorisation

Groupe	Titre Unité de conformité / *Profil*	Facultative
Sécurité	Configuration du Service d'autorisation Serveur	False

6.6.11 Facette Serveur Service KeyCredential

Le Tableau 32 décrit les détails de la Facette *Serveur Service* KeyCredential. Cette Facette définit la capacité d'interagir avec un *Service* KeyCredential pour obtenir des KeyCredentials (authentifiants principaux). Par exemple, les KeyCredentials sont nécessaires pour accéder à un *Service* d'Autorisation ou à un Courtier. Le *Service* KeyCredential fait normalement partie d'un outil commun à tout le système, comme un GDS qui gère aussi les Applications, les Jetons d'accès et les *Certificats*.

Tableau 32 – Facette Serveur Service KeyCredential

Groupe	Titre Unité de conformité / *Profil*	Facultative
Sécurité	Modèle Push pour le *Service* KeyCredential	False

6.6.12 Facette Serveur Attribut WriteMask

Le Tableau 33 décrit les détails de la Facette *Serveur Attribut* WriteMask. Cette Facette définit la capacité de mettre à jour les caractéristiques des *Nœuds* individuels dans l'*AddressSpace* en autorisant l'écriture d'*Attributs de Nœuds*. La Facette exige la prise en charge de l'authentification de l'accès de l'utilisateur, ainsi que l'apport d'informations relatives aux droits d'accès dans l'*AddressSpace* et la limitation réelle des droits d'accès, comme décrit.

Tableau 33 – Facette Serveur Attribut WriteMask

Groupe	Titre Unité de conformité / *Profil*	Facultative
Profil	Sécurité utilisateur Contrôle accès de base	False
Modèle d'espace d'adressage	Espace d'adressage – UserWriteMask	False
Modèle d'espace d'adressage	Espace d'adressage – UserWriteMask Multiniveaux	True
Modèle d'espace d'adressage	Espace d'adressage – WriteMask	False

6.6.13 Facette Serveur Accès fichier

Le Tableau 34 décrit les détails de la Facette *Serveur* Accès fichier. Cette Facette spécifie la prise en charge de la présentation des informations du Fichier via le Type de Fichier défini. Cela comprend la lecture du fichier ainsi que l'écriture facultative des données du fichier.

Tableau 34 – Facette Serveur Accès fichier

Groupe	Titre Unité de conformité / *Profil*	Facultative
Informations de base	Informations de base – FileType Base	False
Informations de base	Informations de base – FileType Ecriture	True

6.6.14 Facette Serveur Documentation

Le Tableau 35 décrit les détails de la Facette *Serveur* Documentation. Cette Facette définit une liste de documentation utilisateur qu'il convient qu'une application serveur fournisse.

Tableau 35 – Facette Serveur Documentation

Groupe	Titre Unité de conformité / *Profil*	Facultative
Divers	Documentation – Installation	False
Divers	Documentation – Langages multiples	True
Divers	Documentation – En ligne	True
Divers	Documentation – *Profils* pris en charge	True
Divers	Documentation – Guide de Recherche de pannes	True
Divers	Documentation – Guide Utilisateur	False

6.6.15 Facette Serveur Abonnement intégré aux DataChanges

Le Tableau 36 décrit les détails de la Facette *Serveur Abonnement* intégré aux DataChanges. Cette Facette spécifie le niveau minimal de prise en charge des notifications de modification de données au sein des abonnements. Elle inclut les limites permettant de réduire le plus possible la charge de mémoire et de traitement nécessaire à la mise en œuvre de la facette. Cette Facette comprend la fonctionnalité de création, modification et suppression des Abonnements, ainsi que d'ajout, de modification et de suppression des Eléments Surveillés. Pour chaque *Session*, les *Serveurs* doivent au minimum prendre en charge un *Abonnement* comportant jusqu'à deux éléments. De plus, il est nécessaire de prendre en charge deux requêtes d'édition en parallèle. Cette Facette est disponible pour une plateforme telle que celle fournie par le *Profil Serveur* à dispositif micro-intégré dont la mémoire est limitée et nécessite d'être gérée.

Tableau 36 – Facette Serveur Abonnement intégré aux DataChanges

Groupe	Titre Unité de conformité / *Profil*	Facultative
Services Eléments surveillés	Contrôle de base	False
Services Eléments surveillés	Contrôle 2 éléments	False
Services Eléments surveillés	Contrôle QueueSize_1	False
Services Eléments surveillés	Contrôle Modification de valeur	False
Services Abonnement	*Abonnement* de base	False
Services Abonnement	*Abonnement* Minimum 1	False
Services Abonnement	*Abonnement* Politique d'élimination d'édition	False
Services Abonnement	*Abonnement* Edition Min 02	False

6.6.16 Facette Serveur Abonnement normalisé aux DataChanges

La Facette Serveur Abonnement normalisé aux DataChanges n'est plus conseillée, car elle est annulée et remplacée par la Facette Serveur 2017 Abonnement normalisé aux DataChanges dans la v1.04.

6.6.17 Facette Serveur 2017 Abonnement normalisé aux DataChanges

Le Tableau 37 décrit les détails de la Facette *Serveur* 2017 *Abonnement* normalisé aux DataChanges. Cette Facette spécifie la prise en charge normalisée de l'abonnement aux modifications de données et étend les caractéristiques et les limites définies par la Facette *Abonnement* intégré aux Modifications de données. Voir ces limites dans les *ConformanceUnits* (Unités de conformité). Noter que la *Méthode* Appel Service est exigée seulement pour les *Méthodes* définies dans cette Facette. Cette Facette annule et remplace la "Facette *Serveur Abonnement* normalisé aux DataChanges".

Tableau 37 – Facette Serveur 2017 Abonnement normalisé aux DataChanges

Groupe	Titre Unité de conformité / *Profil*	Facultative
Profil	Facette Serveur Abonnement intégré aux DataChanges	False
Informations de base	Informations de base – *Méthode* GetMonitoredItems	False
Informations de base	Informations de base – *Méthode* ResendData	False
Services Méthode	*Méthode* Appel	False
Services Eléments surveillés	Contrôle 10 éléments	False
Services Eléments surveillés	Contrôle 100 éléments	False
Services Eléments surveillés	Contrôle MinQueueSize_02	False
Services Eléments surveillés	Contrôle Déclenchement	False
Services Eléments surveillés	Filtre de Bande Morte d'Eléments Surveillés	False
Services Abonnement	*Abonnement* Minimum 02	False
Services Abonnement	*Abonnement* Edition Min 05	False

6.6.18 Facette Serveur Abonnement amélioré aux DataChanges

La Facette *Serveur Abonnement* amélioré aux DataChanges n'est plus conseillée, car elle est annulée et remplacée par la Facette Serveur 2017 Abonnement amélioré aux DataChanges dans la v1.04.

6.6.19 Facette Serveur 2017 Abonnement amélioré aux DataChanges

Le Tableau 38 décrit les détails de la Facette *Serveur* 2017 *Abonnement* amélioré aux DataChanges. Cette Facette spécifie une prise en charge améliorée de l'abonnement aux modifications de données. Elle fait partie du *Profil* 2017 *Serveur* UA normalisé. Cette Facette augmente les limites définies par la Facette *Serveur* 2017 *Abonnement* normalisé aux DataChanges.

Tableau 38 – Facette Serveur 2017 Abonnement amélioré aux DataChanges

Groupe	Titre Unité de conformité / *Profil*	Facultative
Profil	Facette *Serveur* 2017 *Abonnement* normalisé aux DataChanges	False
Services Eléments surveillés	Contrôle 500 éléments	False
Services Eléments surveillés	Contrôle MinQueueSize_05	False
Services Abonnement	*Abonnement* Minimum 05	False
Services Abonnement	*Abonnement* Edition Min 10	False

6.6.20 Facette Serveur Abonnement durable

Le Tableau 39 décrit les détails de la Facette *Serveur Abonnement* durable. Cette Facette spécifie la prise en charge du stockage durable des données et des événements, même lorsque les *Clients* sont déconnectés. Cette Facette implique la prise en charge de l'une des Facettes *Abonnement* aux DataChange ou aux *Evénements*.

Tableau 39 – Facette Serveur Abonnement durable

Groupe	Titre Unité de conformité / *Profil*	Facultative
Services Abonnement	*Abonnement* Durable	False
Services Abonnement	*Abonnement* Durable StorageLevel Elevé	True
Services Abonnement	*Abonnement* Durable StorageLevel Moyen	True
Services Abonnement	*Abonnement* Durable StorageLevel Faible	True

6.6.21 Facette Serveur Accès aux données

Le Tableau 40 décrit les détails de la Facette *Serveur* Accès aux Données. Cette Facette spécifie la prise en charge d'un Modèle d'information utilisé pour fournir des données d'automatisation industrielle. Ce modèle définit des structures normalisées pour les éléments de données analogiques et discrets, et leur qualité de service. Cette Facette étend la Facette *Serveur* principal qui inclut la prise en charge du comportement de l'*AddressSpace* de base.

Tableau 40 – Facette Serveur Accès aux données

Groupe	Titre Unité de conformité / *Profil*	Facultative
Accès aux données	Accès aux données – AnalogItems	True
Accès aux données	Accès aux données – ArrayItemType	True
Accès aux données	Accès aux données – Nombre Complexe	True
Accès aux données	Accès aux données – DataItems	False
Accès aux données	Accès aux données – Nombre DoubleComplex	True
Accès aux données	Accès aux données – MultiState	True
Accès aux données	Accès aux données – MultiStateValueDiscrete	True
Accès aux données	Accès aux données – PercentDeadband	True
Accès aux données	Accès aux données – SemanticChanges	True
Accès aux données	Accès aux données – TwoState	True

6.6.22 Facette Serveur ComplexType

La Facette Serveur ComplexType n'est plus conseillée, car elle est annulée et remplacée par la Facette Serveur 2017 ComplexType dans la v1.04.

6.6.23 Facette Serveur 2017 ComplexType

Le Tableau 41 décrit les détails de la Facette *Serveur* 2017 ComplexType. Cette Facette étend la Facette *Serveur* principal afin d'inclure des *Variables* avec des données structurées, c'est-à-dire des données composées d'éléments multiples tels qu'une structure et dans lesquelles les éléments individuels sont présentés comme variables de composant. La prise en charge de cette Facette exige la mise en œuvre de DataTypes structurées et de *Variables* qui utilisent ces DataTypes. Le *Jeu de services* Read, Write et Subscriptions doit prendre en charge le codage et le décodage de ces DataTypes structurés. Le *Serveur* peut également prendre en charge d'autres codages (tels qu'un codage XML) lorsque le protocole binaire est utilisé et inversement.

Tableau 41 – Facette Serveur 2017 ComplexType

Groupe	Titre Unité de conformité / *Profil*	Facultative
Modèle d'espace d'adressage	Espace d'adressage – Attribut DataTypeDefinition	False
Services Attribut	*Attribut* Codage alternatif	True
Services Attribut	*Attribut* Lecture Complexe	False
Services Attribut	*Attribut* Ecriture Complexe	False
Services Eléments surveillés	Contrôle Codage alternatif	True
Services Eléments surveillés	Contrôle Valeur Complexe	True

6.6.24 Facette Serveur Abonnement normalisé aux événements

Le Tableau 42 décrit les détails de la Facette *Serveur Abonnement normalisé aux événements*. Cette Facette spécifie la prise en charge normalisée d'abonnement aux événements et est destinée à compléter les *Profils* Complets. La prise en charge de cette Facette exige la mise en œuvre des *Types d'événements* qui représentent les *Evénements* que le *Serveur* peut signaler, ainsi que leurs champs spécifiques. Elle exige également qu'au moins l'*Objet Serveur* comporte l'ensemble d'*Attributs EventNotifier*. La prise en charge inclut les *Services* de création, modification et suppression des Abonnements, ainsi que d'ajout, de modification et de suppression des Eléments Surveillés pour les *Nœuds* d'*Objet* avec un "*Attribut EventNotifier*". La création d'un élément de surveillance peut inclure un filtre qui comporte des Opérandes de Filtrage Attributs Simples et une liste sélective d'Opérateurs. Les

Opérateurs sont les suivants: Equals, IsNull, GreaterThan, LessThan, GreaterThanOrEqual, LessThanOrEqual, Like, Not, Between, InList, And, Or, Cast, BitwiseAnd, BitwiseOr et TypeOf. La prise en charge de filtres plus complexes est facultative. Cette Facette a été mise à jour pour inclure plusieurs *ConformanceUnits* facultatives d'informations de base. Ces *ConformanceUnits* sont facultatives pour assurer la rétrocompatibilité. A l'avenir, ces *ConformanceUnits* facultatives deviendront obligatoires et il est donc fortement recommandé que tous les serveurs les prennent en charge.

Tableau 42 – Facette Serveur Abonnement normalisé aux événements

Groupe	Titre Unité de conformité / *Profil*	Facultative
Modèle d'espace d'adressage	Espace d'adressage – Evénements	False
Informations de base	Informations de base – Défaillance d'appareil	True
Informations de base	Informations de base – EventType EventQueueOverflow	True
Informations de base	Informations de base – Evénements de progression	True
Informations de base	Informations de base – SemanticChange	True
Informations de base	Informations de base – Statut Système	True
Informations de base	Informations de base – Statut Système – Système sous-jacent	True
Services Eléments surveillés	Contrôle de base	False
Services Eléments surveillés	Contrôle Filtre d'*Evénements* complexes	True
Services Eléments surveillés	Contrôle d'Evénements	False
Services Eléments surveillés	Contrôle 10 éléments	False
Services Eléments surveillés	Contrôle QueueSize_ServerMax	False
Services Abonnement	*Abonnement* de base	False
Services Abonnement	*Abonnement* Minimum 02	False
Services Abonnement	*Abonnement* Politique d'élimination d'édition	False
Services Abonnement	*Abonnement* Edition Min 05	False

6.6.25 Facette Serveur Notification de l'espace d'adressage

Le Tableau 43 décrit les détails de la Facette *Serveur* Notification de l'espace d'adressage. Cette Facette exige la prise en charge d'une hiérarchie de *Nœuds* d'*Objet* qui sont des notifications et de *Nœuds* qui sont des sources d'événement. La hiérarchie est utilisée couramment comme méthode d'organisation d'une installation en zones qui peuvent être gérées par différents opérateurs.

Tableau 43 – Facette Serveur Notification de l'espace d'adressage

Groupe	Titre Unité de conformité / *Profil*	Facultative
Modèle d'espace d'adressage	Espace d'adressage – Hiérarchie de Notification	False
Modèle d'espace d'adressage	Espace d'adressage – Hiérarchie de Source	False

6.6.26 Facette Serveur A & C Condition de base

Le Tableau 44 décrit les détails de la Facette *Serveur* A & C *Condition* de base. Cette Facette exige la prise en charge de base pour les *Conditions*. Les Informations concernant les *Conditions* sont fournies au moyen des Notifications d'*Evénements* et cette Facette est donc fondée sur la Facette *Serveur Abonnement normalisé aux événements*. Les *Conditions* dont l'état est "intéressant" (selon la définition du *Serveur*) peuvent être rafraîchies à l'aide de la *Méthode* de rafraîchissement qui exige la prise en charge de la Facette *Serveur Méthode*. Le

Serveur peut également, de manière facultative, prévoir la prise en charge des classes de *Conditions*.

Tableau 44 – Facette Serveur A & C Condition de base

Groupe	Titre Unité de conformité / *Profil*	Facultative
Profil	Facette *Serveur Méthode*	False
Profil	Facette *Serveur Abonnement normalisé aux événements*	False
Alarmes et *conditions*	A & C – Base	False
Alarmes et *conditions*	A & C – Sous Classes de *Conditions*	True
Alarmes et *conditions*	A & C – ConditionClasses	True
Alarmes et *conditions*	A & C – Rafraîchissement	False

6.6.27 Facette Serveur A & C Refresh2

Le Tableau 45 décrit les détails de la Facette *Serveur* A & C Refresh2. Cette Facette renforce la Facette *Serveur* A & C *Condition* de base par la prise en charge de la *Méthode* ConditionRefresh2.

Tableau 45 – Facette Serveur A & C Refresh2

Groupe	Titre Unité de conformité / *Profil*	Facultative
Profil	Facette Serveur A & C *Condition* de base	False
Alarmes et *conditions*	A & C – Refresh2	False

6.6.28 Facette Serveur A & C Instance de l'espace d'adressage

Le Tableau 46 décrit les détails de la Facette *Serveur* A & C Instance Espace d'adressage. Cette Facette spécifie la prise en charge nécessaire pour qu'un *Serveur* présente les *Alarmes* et *Conditions* dans son *AddressSpace*. Cela inclut le modèle d'information A & C *AddressSpace*.

Tableau 46 – Facette Serveur A & C Instance Espace d'adressage

Groupe	Titre Unité de conformité / *Profil*	Facultative
Alarmes et *conditions*	A & C – Instances	False

6.6.29 Facette Serveur A & C Activation

Le Tableau 47 décrit les détails de la Facette *Serveur* A & C Activer. Cette Facette exige l'activation et la désactivation des *Conditions*. Cette Facette s'établit sur la Facette *Serveur A & C Condition* de base. L'activation et la désactivation exigent également l'existence d'instances de ces ConditionTypes dans l'AddressSpace dans la mesure où la *Méthode* Activer ne peut être invoquée que sur une instance de la *Condition*.

Tableau 47 – Facette Serveur A & C Activer

Groupe	Titre Unité de conformité / *Profil*	Facultative
Profil	Facette Serveur A & C *Condition* de base	False
Alarmes et *conditions*	A & C – Activer	False
Alarmes et *conditions*	A & C – Instances	False

6.6.30 Facette Serveur A & C AlarmMetrics

Le Tableau 48 décrit les détails de la Facette *Serveur* A & C AlarmMetrics. Cette Facette exige la prise en charge d'AlarmMetrics. AlarmMetrics présente l'état et les éventuels problèmes dans le système d'alarme. Un *Serveur* peut fournir ces paramètres à plusieurs niveaux (poste de l'opérateur, zone de l'installation, système global, etc.).

Tableau 48 – Facette Serveur A & C AlarmMetrics

Groupe	Titre Unité de conformité / *Profil*	Facultative
Alarmes et *conditions*	A & C – Paramètres d'*Alarme*	False

6.6.31 Facette Serveur A & C Alarme

Le Tableau 49 décrit les détails de la Facette *Serveur* A & C *Alarme*. Cette Facette exige la prise en charge des *Alarmes*. Les Alarmes étendent le Type de Condition par l'ajout d'un état Actif qui indique le moment où un élément constitutif du système exige l'attention d'un Opérateur. Cette Facette s'établit sur la Facette *Serveur A & C Condition* de base. Elle permet également la prise en charge facultative de l'ordonnancement, des commentaires d'alarme et autres types d'alarmes discrètes tels que Trip (Déclenchement) ou Off-Normal (Non Normalisé).

Tableau 49 – Facette Serveur A & C Alarme

Groupe	Titre Unité de conformité / *Profil*	Facultative
Profil	Facette Serveur A & C *Condition* de base	False
Alarmes et *conditions*	A & C – *Alarme*	False
Alarmes et *conditions*	A & C – Son Audible	True
Alarmes et *conditions*	A & C – Commentaire	True
Alarmes et *conditions*	A & C – Anomalie	True
Alarmes et *conditions*	A & C – Discrètes	False
Alarmes et *conditions*	A & C – Alarme "First in Group"	True
Alarmes et *conditions*	A & C – OffNormal	True
Alarmes et *conditions*	A & C – Activation et désactivation différées	True
Alarmes et *conditions*	A & C – Hors Service	True
Alarmes et *conditions*	A & C – Répétition des alarmes	True
Alarmes et *conditions*	A & C – Suspension	True
Alarmes et *conditions*	A & C – Silence	True
Alarmes et *conditions*	A & C – Suppression	True
Alarmes et *conditions*	A & C – Suppression par l'opérateur	True
Alarmes et *conditions*	A & C – OffNormal	True
Alarmes et *conditions*	A & C – Déclenchement	True

6.6.32 Facette Serveur A & C Alarme acquittable

Le Tableau 50 décrit les détails de la Facette *Serveur* A & C *Alarme* acquittable. Cette Facette exige la prise en charge de l'acquittement des *Alarmes* actives. Cette Facette est fondée sur la Facette *Serveur* A & C *Alarme*. L'acquittement exige la prise en charge de la *Méthode* Acknowledge et de l'état Acquitté (Acknowledged). La prise en charge de l'état Confirmé et de la *Méthode* Confirmation est facultative.

Tableau 50 – Facette Serveur A & C Alarme acceptable

Groupe	Titre Unité de conformité / *Profil*	Facultative
Profil	Facette Serveur A & C Alarme	False
Alarmes et *conditions*	A & C – Acquittement	False
Alarmes et *conditions*	A & C – Confirmer	True

6.6.33 Facette Serveur A & C Alarme exclusive

Le Tableau 51 décrit les détails de la Facette *Serveur* A & C Alarme exclusive. Cette Facette exige la prise en charge des *Alarmes* ayant des sous-états multiples, qui identifient les différentes *Conditions* limites. Cette Facette est fondée sur la Facette *Serveur* A & C *Alarme*. Le terme "exclusive" signifie qu'un seul sous-état peut être actif au même moment. Par exemple, si une température est supérieure à la limite très élevée (HighHigh), l'alarme de niveau exclusive associée présente alors un sous-état de niveau très élevé et non un sous-état de niveau élevé (High). Cette Facette exige d'un *Serveur* qu'il prenne en charge au moins un des modèles d'*Alarme* facultatifs: Limite, RateOfChange ou Ecart.

Tableau 51 – Facette Serveur A & C Alarme exclusive

Groupe	Titre Unité de conformité / *Profil*	Facultative
Profil	Facette Serveur A & C Alarme	False
Alarmes et *conditions*	A & C – Ecart exclusif	True
Alarmes et *conditions*	A & C – Niveau exclusif	True
Alarmes et *conditions*	A & C – Limite exclusive	False
Alarmes et *conditions*	A & C – RateOfChange exclusif	True

6.6.34 Facette Serveur A & C Alarme non exclusive

Le Tableau 52 décrit les détails de la Facette *Serveur* A & C Alarme Non Exclusive. Cette Facette exige la prise en charge des *Alarmes* ayant des sous-états multiples, qui identifient les différentes *Conditions* limites. Cette Facette est fondée sur la Facette *Serveur* A & C *Alarme*. Le terme "non exclusive" signifie que plus d'un sous-état peut être actif au même moment. Par exemple, si une température est supérieure à la limite très élevée (HighHigh), l'alarme de niveau non exclusive associée présente à la fois un sous-état de niveau élevé et un sous-état de niveau très élevé. Cette Facette exige d'un Serveur qu'il prenne en charge au moins un des modèles d'alarme facultatifs: Limite, RateOfChange ou Ecart.

Tableau 52 – Facette Serveur A & C Alarme non exclusive

Groupe	Titre Unité de conformité / *Profil*	Facultative
Profil	Facette Serveur A & C Alarme	False
Alarmes et *conditions*	A & C – Ecart non exclusif	True
Alarmes et *conditions*	A & C – Niveau non exclusif	True
Alarmes et *conditions*	A & C – Limite non exclusive	False
Alarmes et *conditions*	A & C – RateOfChange non exclusif	True