IEC 62541-7

Edition 2.0 2015-03

INTERNATIONAL STANDARD

NORME INTERNATIONALE

colour inside

OPC unified architecture – Part 7: Profiles

Architecture unifiée OPC – Partie 7: Profils

IEC 62541-7:2015-03(en-fr)

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

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

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

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

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

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 It also gives information on projects, replaced and withdrawn publications.

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

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

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

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

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

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

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

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

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

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

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

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

IEC 62541-7

Edition 2.0 2015-03

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

ISBN 978-2-8322-2372-7

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

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

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

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

b) New Security Policy for asymmetric key length > 2048

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

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

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

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

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

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

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

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

OPC UNIFIED ARCHITECTURE –

Part 7: Profiles

1 Scope

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

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

2 Normative references

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

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

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

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

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

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

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

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

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

IEC 62541-11[1], *OPC unified architecture – Part 11: Historical access*

IEC 62541-13[1], *OPC unified architecture – Part 13: Aggregates*

[1] To be published.

3 Terms, definitions, and conventions

3.1 Terms and definitions

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

3.1.1
application
software program that executes or implements some aspect of OPC UA

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

3.1.2
ConformanceUnit
specific set of OPC UA features that can be tested as a single entity

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

3.1.3
ConformanceGroup
group of *ConformanceUnits* that is given a name

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

3.1.4
Facet
Profile dedicated to a specific feature that a *Server* or *Client* may require

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

3.1.5
FullFeatured Profile
Profile that defines all features necessary to build a functional OPC UA *Application*

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

3.1.6
ProfileCategory
arranges *Profiles* into application classes, such as *Server* or *Client*

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

3.1.7
TestCase
technical description of a set of steps required to test a particular function or information model

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

3.1.8
TestLab
facility that is designated to provide testing services

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

3.2 Abbreviations

DA Data Access

HA Historical Access

HMI Human Machine Interface

NIST National Institute of Standard and Technology

PKI Public Key Infrastructure

RSA Rivest-Shamir-Adleman

UA Unified Architecture

4 Overview

4.1 General

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

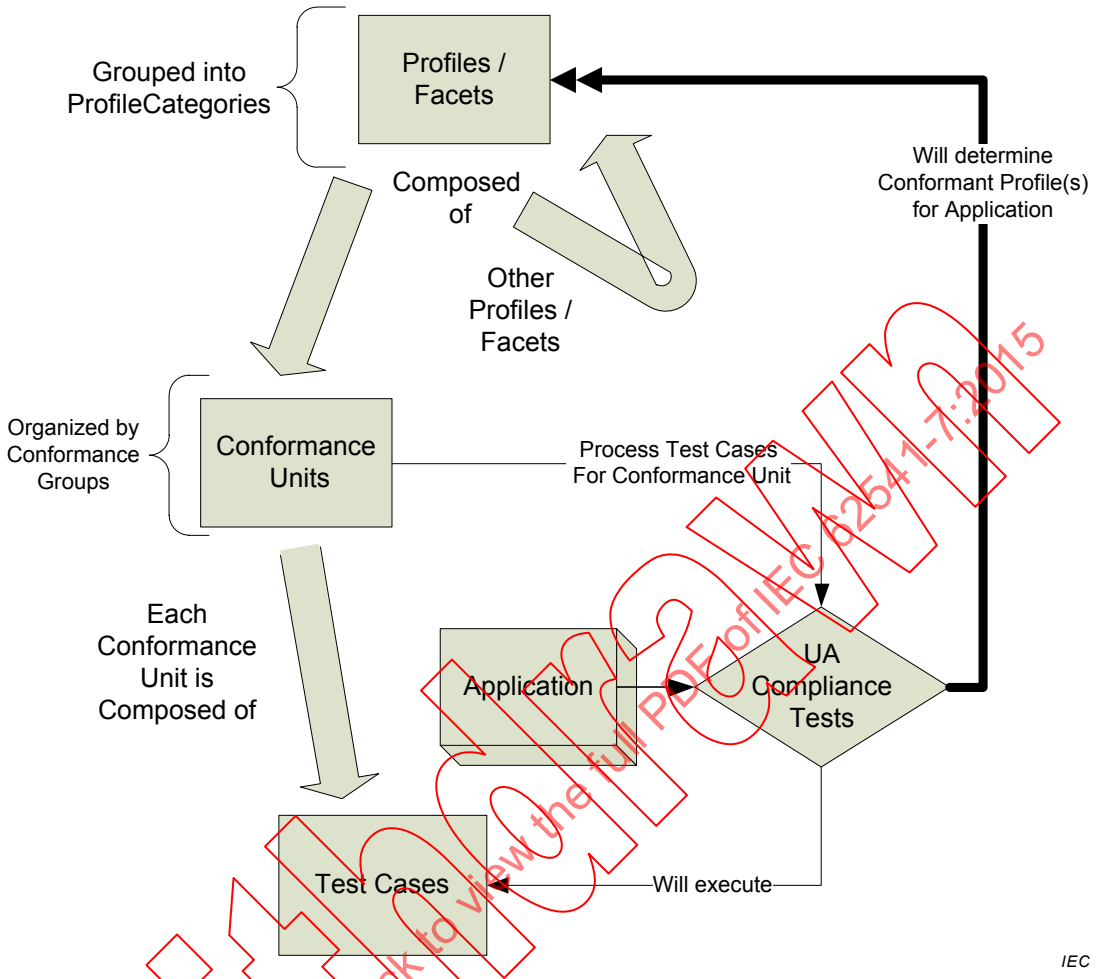


Figure 1 – Profile – ConformanceUnit – TestCases

4.2 ConformanceUnit

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

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

4.3 Profiles

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

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

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

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

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

4.4 Profile Categories

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

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

Table 1 – ProfileCategories

Category	Description
Client	*Profiles* of this category specify functions of an OPC UA *Client*. The URI of such *Profiles* can be part of a Software *Certificate* passed in the ActivateSession request.
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 *Servers* and *Clients*.
Server	*Profiles* of this category specify functions of an OPC UA *Server*. The URI of such *Profiles* can be part of a Software *Certificate* returned with the CreateSession service response and 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 *Servers* and *Clients*.

5 ConformanceUnits

5.1 Overview

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

Table 2 – ConformanceGroups

Group	Description
Address Space Model	Defines *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*.
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.
Query Services	A Query may be used to provide advanced filtering and return a subset of data.
Redundancy	The design of OPC UA ensures that vendors can create redundant *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

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

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

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

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

Table 3 – Discovery Services

Category	Title	Description	Derived
Server	Discovery Get Endpoints	Support the GetEndpoints *Service* to obtain all Endpoints of the *Server*. This includes filtering based on *Profiles*.	
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 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 Get Endpoints Basic	Uses the GetEndpoints *Service* to obtain all Endpoints for a given *Server* URI.	
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	Derived
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*).	
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.	
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.	

Category	Title	Description	Derived
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	Client Entry-Level Support	The *Client* is able to interoperate with *Servers* with lowest level functionality. This includes the ability to operate with a single *Session*, a pre-knowledge of the OPC UA Types (the *Server* may not expose them in the *AddressSpace*), and the ability to use Read vs. *Subscriptions* for monitoring. There may be further restrictions provided by the *Server* via the *Server* capabilities.	

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

Table 5 – Node Management Services

Category	Title	Description	Derived
Server	Node Management Add Node	Support the AddNodes *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	Derived
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 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 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	Derived
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.	
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 must 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 must 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 Read with proper Encoding	This *ConformanceUnit* refers to the ability of a *Server* to support more than one Data Encoding for *Attribute* values. *Clients* can discover the available encodings and can explicitly choose one when calling the Read *Service*.	
Client	Attribute Client Read Complex	Read and decode Values with Structured DataTypes.	

Category	Title	Description	Derived
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.	
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 must be provided: InsertData, InsertEvent, ReplaceData, ReplaceEvent, 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	Derived
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	Derived
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	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*.	
Server	Monitor Items 10	Support at least 10 *MonitoredItems* per *Subscription*.	
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*.	

Category	Title	Description	Derived
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 complex *Event* filters, where complex is defined as supporting the complex filter operator (TypeOf).	
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 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*.	

Category	Title	Description	Derived
Client	Monitor Client Complex Event Filter	Uses complex *Event* filters.	
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	Derived
Server	Subscription Basic	Support the following *Subscription Services*: CreateSubscription, ModifySubscription, DeleteSubscriptions, Publish, Republish and SetPublishingMode.	
Server	Subscription Minimum 1	Support at least 1 Subscriptions 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 republish is optional and no notification retransmission queue has to be provided however the republish service must be provided and will return the appropriate operation level results.	
Server	Subscription Publish Min 05	Support at least 5 Publish *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.	

Category	Title	Description	Derived
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 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.	

5.3 Transport and communication related features

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

Table 11 – Security

Category	Title	Description	Derived
Security	Security Certificate Validation	A certificate will be validated as specified in IEC 62541-4. This includes among others structure and signature examination. Allowing for some validation errors to be suppressed by administration directive.	
Security	Security None	A suite of algorithms that does NOT provide any security settings: -> SymmetricSignatureAlgorithm – Not Used -> SymmetricEncryptionAlgorithm – Not Used -> AsymmetricSignatureAlgorithm – Not Used -> SymmetricKeyWrapAlgorithm – Not Used -> AsymmetricEncryptionAlgorithm – Not Used -> KeyDerivationAlgorithm – Not Used -> DerivedSignatureKeyLength – 0 The use of this suite of algorithms must be able to be enabled or disabled by an administrator.	
Security	Security 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.	

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

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

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

Category	Title	Description	Derived
Security	Security TLS_RSA_WITH_RC4_128_SHA	The connection is established using TLS_RSA_WITH_RC4_128_SHA. The application needs to be configured to prevent the use of AES based protocol suites (TLS 1.0).	
Security	Security TLS_RSA_WITH_AES_256_CBC_SHA256	The connection is established using TLS_RSA_WITH_AES_256_CBC_SHA 256. That has a MinAsymmetricKeyLength – 2048, MaxAsymmetricKeyLength – 4096, AsymmetricSignatureAlgorthm – RSA_SHA256. (TLS 1.2)	
Security	Security Encryption Required	Encryption is required using the algorithms provide in the security algorithm suite.	
Security	Security Signing Required	Signing is required using the algorithms provide in the security algorithm suite.	
Security	Security Time Synch – Configuration	Application supports configuring acceptable clock skew.	
Security	Security Time Synch – NTP / OS Based support	Application supports time synchronization, either via an implementation of Network Time Protocol (NTP), or via features of a standard operating system.	
Security	Security Time Synch – UA based support	An application makes use of the responses header timestamp provided by a configured well know source, such as a *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. * select the allowed User identification policy or policies (User Name/Password or X509 or Kerberos or Anonymous). * enable/disable the security policy "None" or other security policies. * enable/disable endpoints with MessageSecurityMode SIGN or SIGNANDENCRYPT. * set the permitted certification authorities. * define how to react to unknown *Certificates*.	
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).	

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

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

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

Table 12 – Protocol and Encoding

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

5.4 Information Model and AddressSpace related features

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

Table 13 – Base information

Category	Title	Description	Derived
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, than the actual events are verified.	
Server	Base Info Diagnostics	The *Server* supports Diagnostic *Objects* and *Variables*.	
Server	Base Info System Status	The *Server* supports generating SystemStatusChangeEventType indicating shutdown of the *Server* (SourceNode=*Server*).	
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 GetMonitoredItems Method	The *Server* supports obtaining subscription information via GetMonitoredItems *Method* on the *Server* object.	
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.	

Category	Title	Description	Derived
Server	Base Info Custom Type System	The *Server* supports defining user defined *ObjectTypes*, VariableTypes, *ReferenceType* and DataTypes. Supporting this conformance unit does not require that a *Server* exposes the OPC UA *Object*, *Variable*, *Reference*, or Data Types, unless the *Server* implements User types. If User types are defined than the full type-hierarchy has to be exposed as well.	
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 EventQueueOverflowEventType	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.	
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*.	

Category	Title	Description	Derived
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 System Status	The *Client* makes use of SystemStatusChangeEventType to detect server shutdowns.	
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 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 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.	

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

Table 14 – Address Space model

Category	Title	Description	Derived
Server	Address Space Base	Support the *NodeClasses* with their *Attributes* and behaviour as defined in IEC 62541-3. This includes for instance: *Object*, *ObjectType*, *Variable*, *VariableType*, *References* and DataType.	
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 DataTypes	Support StructuredDataTypes with a Data Dictionary.	
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*.	
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.	

Category	Title	Description	Derived
Client	Address Space Client Complex DataTypes	Uses and understands arbitrary StructuredDataTypes via Data Dictionary.	
Client	Address Space Client Notifier Hierarchy	Uses hierarchy of *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. *Server* could expose this information model and *Client* could utilize this information model.

Table 15 – Data Access

Category	Title	Description	Derived
Server	Data Access DataItems	Provide *Variables* of DataItemType or one of its subtypes. Support the StatusCodes specified in the 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 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.	
Client	Data Access Client Basic	Understand the DataAccess *Variable* Types. Make use of the standard Properties if applicable.	
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	Derived
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 Instances	Support the exposing of A&C *Conditions* in the *AddressSpace*.	
Server	A & C ConditionClasses	Supports multiple *Condition* classes for grouping and filtering of *Alarms*.	
Server	A & C Acknowledge	Support Acknowledge, includes Acknowledge *Method*, Acknowledgeable type.	
Server	A & C Confirm	Support confirming *Conditions*, includes Confirm method.	
Server	A & C Comment	Support Comments, includes AddComment *Method*.	
Server	A & C Alarm	Support for Basic *Alarm* functionality, including active, inactive states.	
Server	A & C Branch	Support for *Alarm* Branches which includes previous *Condition* Instances, i.e. conditions instance other than the current condition that still requires some operator action, such as acknowledgement or a dialog.	
Server	A & C Shelving	Support for the shelving mode, including the TimedShelve, OneShotShelve and Unshelve methods.	
Server	A & C Exclusive Level	Supports Exclusive Level *Alarm* type.	
Server	A & C Exclusive Limit	Supports Exclusive Limit *Alarms*. A *Server* that supports this must support 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 must support 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 Off Normal	Supports Off Normal *Alarm* type.	
Server	A & C Trip	Supports Trip *Alarm* type.	
Server	A & C Dialog	Supports DialogConditionType including Respond *Method*.	
Server	A & E Wrapper Mapping	The *Server* uses the COM A&E mapping specified in IEC 62541-9 to map COM Events to A&C Events, this includes *Condition* Class mapping.	

Category	Title	Description	Derived
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 Instances Client	Uses A&C *Conditions* that are exposed in the *AddressSpace*.	
Client	A & C ConditionClasses Client	Uses *Condition* classes to group *Alarms*.	
Client	A & C Acknowledge Client	Uses Acknowledge, including Acknowledge *Method*, Acknowledgeable type.	
Client	A & C Confirm Client	Uses confirming *Conditions*, including Confirm method.	
Client	A & C Comment Client	Uses Comments, including AddComment *Method*.	
Client	A & C Alarm Client	Uses Basic *Alarm* functionality, including active, inactive states.	
Client	A & C Branch Client	Uses *Alarm* Branches which included previous *Condition* Instances, i.e. conditions instance other than the current condition that still requires some action, such as acknowledgement or confirmation.	
Client	A & C Shelving Client	Uses the shelving model, including the TimedShelve, OneShotShelve and Unshelve methods.	
Client	A & C Exclusive Level Client	Uses Exclusive Level *Alarms* as defined.	
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 Off Normal Client	Uses the Off Normal *Alarm* types.	
Client	A & C Trip Client	Uses the Trip *Alarm* type.	
Client	A & C Dialog Client	Uses the DialogConditionType including Respond *Method*.	

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

Table 17 – Historical Access

Category	Title	Description	Derived
Server	Historical Access Read Raw	General support for basic historical access, reading raw data using the ReadRawModifiedDetails structure. Where the time range is specified using a start time, stop time and number of values (a minimum of two of the three parameters must be provided) and the ReadModified flag is set to False.	
Server	Historical Access Data Max Nodes Read Continuation Point	Supports enough continuation points to cover the number of supported points indicated in the MaxNodesPerHistoryReadData *Server* OperationLimits parameter for historical data access.	
Server	Historical Access Time Instance	Supports reading historical data at a specified instance in time using the ReadAtTimeDetails structure.	
Server	Historical Access Aggregates	Supports reading one or more Aggregates of historical values of *Variables* using the ReadProcessedDetails structure. At least one of the Aggregates described in IEC 62541-13 must be supported. The complete list will be shown in the *Software Certificate*.	
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.	

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

Category	Title	Description	Derived
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. The complete list of Aggregates used by the *Client* is included in the results of this *ConformanceUnit*.	
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	Derived
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	Derived
Server	Aggregate master configuration	Supports at least one master AggregateConfigurationType *Object* as part of the *Server* configuration.	
Server	Aggregate optional configuration	Supports at least one optional AggregateConfigurationType *Object*. Optional AggregateConfigurationType *Objects* occur at different levels from the master AggregateConfigurationType *Object*.	
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.	

Category	Title	Description	Derived
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.	
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.	

Category	Title	Description	Derived
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.	
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.	

Category	Title	Description	Derived
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.	
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.	

Category	Title	Description	Derived
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.	

Category	Title	Description	Derived
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.	
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 NumberOfTransition	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 StandardDevSample	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 StandardDevPopulation	Subscribes for data using the StandardDeviationPopulation Aggregate filter.	

Category	Title	Description	Derived
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	Derived
Server	Auditing Base	Support AuditEvents. The list of supported AuditEvents shall be verified during certification testing and will be shown in the *Software Certificate*. 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	Derived
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.	

5.5 Miscellaneous

The following table describes miscellaneous *ConformanceUnits*.

Each table includes a listing of the *Profile Category* to which a *ConformanceUnit* belongs, the title and description of the *ConformanceUnit* and a column that indicates if the

ConformanceUnit is derived from another *ConformanceUnit*. A *ConformanceUnit* that is derived from another *ConformanceUnit* includes all of the same tests as its parent plus one or more additional TestCases. These TestCases can only further restrict the existing TestCases.

Table 21 – Miscellaneous

Category	Title	Description	Derived
Client, 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.	
Client, Server	Documentation – Multiple Languages	The documentation is available in multiple languages. The results of this conformance unit include the list of supported languages.	
Client, 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.	
Client, 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.	
Client, 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.	
Client, 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.	

6 Profiles

6.1 Overview

Clause 6 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 22 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. This URI is also included in the *SoftwareCertificate* associated with the *Profile*. The URI is case sensitive.

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 support for optional *ConformanceUnits* is described in the certificate that is generated by the associated testing. Optional *ConformanceUnits* are clearly identified in this document and as part of the *SoftwareCertificate* that describes the *Profiles* supported by a product. The *SoftwareCertificate* must show all optional *ConformanceUnits* and if they are support. Any on-line displays that list the *Profiles* a product supports must also include the optional *ConformanceUnits*. Some *ConformanceUnits* also include lists of supported DataTypes or optional Subtypes, the list 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 22 – Profile list

Profile	Related Category	URI
Core Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/CoreFacet
Base Server Behaviour Facet	Server	http://opcfoundation.org/UA-Profile/Server/Behaviour
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 Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/StandardDataChangeSubscription
Enhanced DataChange Subscription Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/EnhancedDataChangeSubscription
Data Access Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/DataAccess
ComplexType Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/ComplexTypes
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 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 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
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
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 Access Structured Data Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/HistoricalStructuredData
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

Profile	Related Category	URI
Historical Annotation Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/HistoricalAnnotation
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 Insert Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/HistoricalDataInsert
Historical Data Delete Server Facet	Server	http://opcfoundation.org/UA-Profile/Server/HistoricalDataDelete
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 Server Profile	Server	http://opcfoundation.org/UA-Profile/Server/NanoEmbeddedDevice
Micro Embedded Device Server Profile	Server	http://opcfoundation.org/UA-Profile/Server/MicroEmbeddedDevice
Embedded UA Server Profile	Server	http://opcfoundation.org/UA-Profile/Server/EmbeddedUA
Standard UA Server Profile	Server	http://opcfoundation.org/UA-Profile/Server/StandardUA
Core Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/Core
Base Client Behaviour Facet	Client	http://opcfoundation.org/UA-Profile/Client/Behaviour
Discovery Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/Discovery
AddressSpace Lookup Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/AddressSpaceLookup
Entry-Level SupportClient Facet	Client	http://opcfoundation.org/UA-Profile/Client/Entry-LevelSupport
Multi-Server Client Connection Facet	Client	http://opcfoundation.org/UA-Profile/Client/MultiServer
File Access Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/FileAccess
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
DataAccess Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/DataAccess
Event Subscriber Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/EventSubscriber
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 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

Profile	Related Category	URI
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
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 & 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
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 Annotation Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalAnnotation
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 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 Insert Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalInsertData
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 Access Modified Data Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalAccessModifiedData
Historical Structured Data AtTime Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalAtTimeStructuredData
Historical Structured Data Access Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalAccessStructuredData
Historical Structured Data Modified Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalModifiedStructuredData
Historical Structured Data Delete Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalDeleteStructuredData
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 Insert Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalInsertStructuredData
Historical Events Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalEvents
Historical Event Update Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalUpdateEvents

Profile	Related Category	URI
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
Historical Event Insert Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalInsertEvents
Aggregate Subscriber Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/AggregateSubscriber
User Token – Anonymous Facet	Security	http://opcfoundation.org/UA-Profile/Security/UserToken/Anonymous
User Token – User Name Password Server Facet	Server, Security	http://opcfoundation.org/UA-Profile/Security/UserToken-Server/UserNamePassword
User Token – X509 Certificate Server Facet	Server, Security	http://opcfoundation.org/UA-Profile/Security/UserToken-Server/X509Certificate
User Token – Issued Token Server Facet	Server, Security	http://opcfoundation.org/UA-Profile/Security/UserToken-Server/IssuedToken
User Token – Issued Token Windows Server Facet	Server, Security	http://opcfoundation.org/UA-Profile/Security/UserToken-Server/IssuedTokenWindows
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
UA-TCP UA-SC UA Binary	Transport	http://opcfoundation.org/UA-Profile/Transport/uatcp-uasc-uabinary
SOAP-HTTP WS-SC UA XML	Transport	http://opcfoundation.org/UA-Profile/Transport/soaphttp-wssc-uaxml
SOAP-HTTP WS-SC UA Binary	Transport	http://opcfoundation.org/UA-Profile/Transport/soaphttp-wssc-uabinary
SOAP-HTTP WS-SC UA XML-UA Binary	Transport	http://opcfoundation.org/UA-Profile/Transport/soaphttp-wssc-uaxml-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
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 – Program Access	Security, Server	http://opcfoundation.org/UA-Profile/Security/BestPracticeProgramAccess
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

Profile	Related Category	URI
Best Practice – Alarm Handling Client	Client, Security	http://opcfoundation.org/UA-Profile/Security/BestPracticeAlarmHandlingClient
Best Practice – Audit Events Client	Client, Security	http://opcfoundation.org/UA-Profile/Security/BestPracticeAuditEventsClient
SecurityPolicy – None	Security	http://opcfoundation.org/UA/SecurityPolicy#None
SecurityPolicy – Basic128Rsa15	Security	http://opcfoundation.org/UA/SecurityPolicy#Basic128Rsa15
SecurityPolicy – Basic256	Security	http://opcfoundation.org/UA/SecurityPolicy#Basic256
SecurityPolicy – Basic256Sha256	Security	http://opcfoundation.org/UA/SecurityPolicy#Basic256Sha256
TransportSecurity – TLS 1.0	Security	http://opcfoundation.org/UA-Profile/TransportSecurity/TLS-1-0
TransportSecurity – TLS 1.1	Security	http://opcfoundation.org/UA-Profile/TransportSecurity/TLS-1-1
TransportSecurity – TLS 1.2	Security	http://opcfoundation.org/UA-Profile/TransportSecurity/TLS-1-2

The contents of each of the listed *Profiles* will be described in a tabular form in a separate section. 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 the appropriate conformance group section.

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 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.

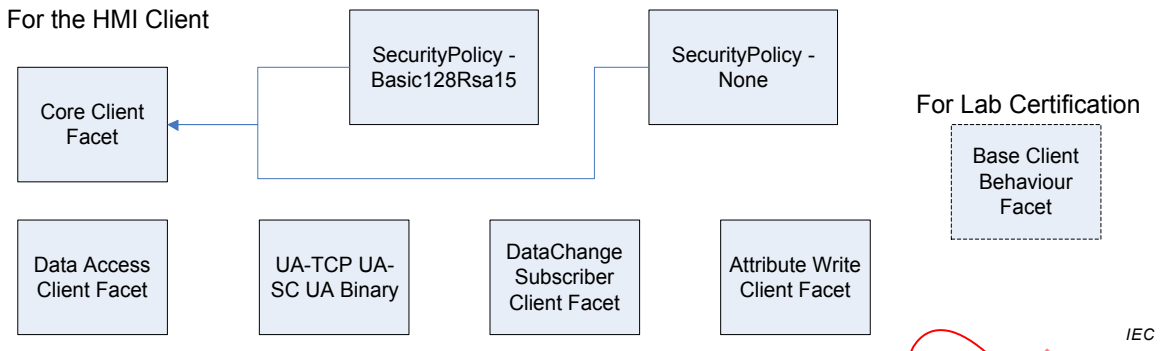


Figure 2 – HMI Client sample

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.

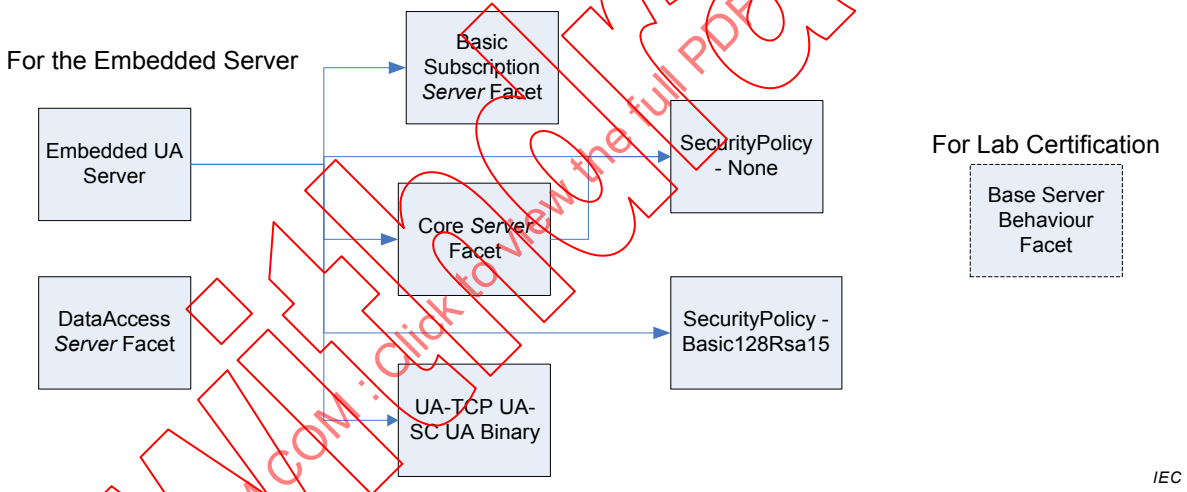


Figure 3 – Embedded Server sample

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.

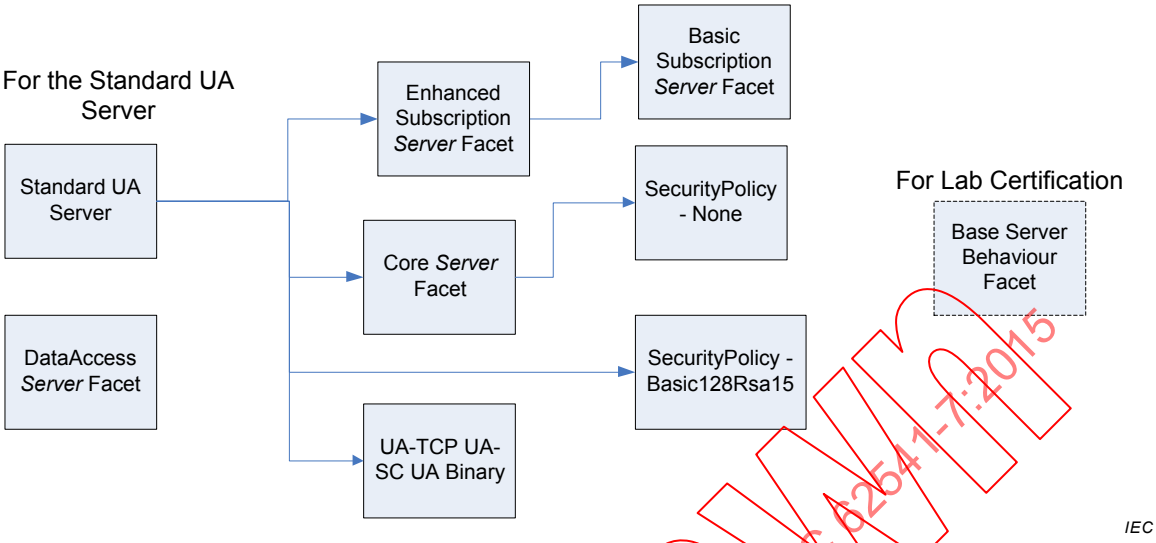


Figure 4 – Standard UA Server sample

If the example HMI *Client* were to connect to either of the example *Servers*, it may have to adjust its behaviour 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.

6.5 Profile tables

6.5.1 Introduction

All subclauses in 6.5 starting with 6.5.2 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 *Profile*. The second column is a brief description of the *ConformanceUnit* or included *Profile*. The last column indicates if the *ConformanceUnit* is optional or required.

6.5.2 Core Server Facet

Table 23 describes the details of the Core *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*, Authentication with UserName and Password. Support for a type system is not required nor does the *Server* need to support encryption and signing of user identity tokens (This assumes the *Server* also supports a transport that provides security.) This Facet has been extended with additional Base Information *ConformanceUnits*. They are optional to provide backward compatibility. In the future the *ConformanceUnit* "Base Info Server Capabilities"will become required, and so it is highly recommended that all *Servers* support it. For broad applicability, it is recommended that *Servers* support multiple transport and security *Profiles*.

Table 23 – Core 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 Base	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 OptionSet	True
Base Information	Base Info Placeholder Modelling Rules	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 – No Application Authentication	True
Security	Security Administration	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

6.5.3 Base Server Behaviour Facet

Table 24 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, to set the security level and to configure the *Discovery Server* and specify where this *Server* shall be registered.

Table 24 – 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

6.5.4 Attribute WriteMask Server Facet

Table 25 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 25 – 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

6.5.5 File Access Server Facet

Table 26 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 26 –File Access Server Facet

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

6.5.6 Documentation Server Facet

Table 27 describes the details of the Documentation *Server* Facet. This Facet defines a list of user documentation that a server application should provide.

Table 27 – 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

6.5.7 Embedded DataChange Subscription Server Facet

Table 28 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, but, republish buffering is not required. 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 28 – 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

6.5.8 Standard DataChange Subscription Server Facet

Table 29 describes the details of the Standard DataChange *Subscription Server* Facet. This Facet specifies the standard support of subscribing to data changes. This Facet extends features and limits defined by the Embedded Data Change *Subscription* Facet. As a minimum, *Servers* shall support 2 Subscriptions with at least 100 items for at least half of the required Sessions. The 100 items shall be supported for at least half of the required Subscriptions. Queuing with up to two queued entries is required. Support of five parallel Publish requests per *Session* is required. This Facet also requires the support of the triggering service. This Facet has been updated to include optional *ConformanceUnits* to allow for backward compatibility. These optional *ConformanceUnits* are highly recommended, in that in a future release they will be made mandatory.

Table 29 – Standard DataChange Subscription Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	Embedded DataChange Subscription Server Facet	False
Base Information	Base Info GetMonitoredItems Method	True
Method Services	Method Call	True
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

6.5.9 Enhanced DataChange Subscription Server Facet

Table 30 describes the details of the Enhanced DataChange *Subscription Server* Facet. This Facet specifies an enhanced support of subscribing to data changes. It is part of the Standard UA *Server Profile*. This Facet increases the limits defined by the Standard Data Change *Subscription* Facet.

Table 30 – Enhanced DataChange Subscription Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	Standard DataChange Subscription 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

6.5.10 Data Access Server Facet

Table 31 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 31 – 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 PercentDeadband	True
Data Access	Data Access Semantic Changes	True
Data Access	Data Access TwoState	True

6.5.11 ComplexType Server Facet

Table 32 describes the details of the ComplexType *Server* Facet. This Facet extends the Core *Server* Facet to include *Variables* with *Complex 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 StructuredDataTypes and *Variables* that make use of these DataTypes. The Read, Write and Subscriptions service set shall support the encoding and decoding of these StructuredDataTypes. 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 32 – ComplexType Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Address Space Model	Address Space Complex DataTypes	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

6.5.12 Standard Event Subscription Server Facet

Table 33 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 33 – Standard Event Subscription Server Facet

Group	Conformance Unit / *Profile* Title	Optional
Address Space Model	Address Space Events	False
Base Information	Base Info EventQueueOverflowEventType	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

6.5.13 Address Space Notifier Server Facet

Table 34 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 34 – 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

6.5.14 A & C Base Condition Server Facet

Table 35 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 35 – 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 ConditionClasses	True
Alarms and Conditions	A & C Refresh	False

6.5.15 A & C Address Space Instance Server Facet

Table 36 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 36 – A & C Address Space Instance Server Facet

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

6.5.16 A & C Enable Server Facet

Table 37 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 37 – 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

6.5.17 A & C Alarm Server Facet

Table 38 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 38 – 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 Comment	True
Alarms and Conditions	A & C Discrete	False
Alarms and Conditions	A & C Off Normal	True
Alarms and Conditions	A & C Shelving	True
Alarms and Conditions	A & C Trip	True

6.5.18 A & C Acknowledgeable Alarm Server Facet

Table 39 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 39 – 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

6.5.19 A & C Exclusive Alarming Server Facet

Table 40 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 40 – 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

6.5.20 A & C Non-Exclusive Alarming Server Facet

Table 41 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 41 – 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

6.5.21 A & C Previous Instances Server Facet

Table 42 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 42 – 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

6.5.22 A & C Dialog Server Facet

Table 43 describes the details of the A & C Dialog *Server* Facet. This Facet requires support of Dialog *Conditions*. This Facet builds upon the A & C Base Condition 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 43 – 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

6.5.23 A & E Wrapper Facet

Table 44 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 44 – 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

6.5.24 Method Server Facet

Table 45 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 45 – Method Server Facet

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

6.5.25 Auditing Server Facet

Table 46 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 must be exposed in the *Server*'s *AddressSpace*

Table 46 – Auditing Server Facet

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

6.5.26 Node Management Server Facet

Table 47 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 47 – 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

6.5.27 Client Redundancy Server Facet

Table 48 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 48 – Client Redundancy Server Facet

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

6.5.28 Redundancy Transparent Server Facet

Table 49 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 49 – Redundancy Transparent Server Facet

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

6.5.29 Redundancy Visible Server Facet

Table 50 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 must 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 50 – Redundancy Visible Server Facet

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

6.5.30 Historical Raw Data Server Facet

Table 51 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 51 – 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

6.5.31 Historical Aggregate Server Facet

Table 52 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 52 – 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 master configuration	False
Aggregates	Aggregate optional configuration	True
Attribute Services	Attribute Historical Read	False
Historical Access	Historical Access Aggregates	False
Historical Access	Historical Access Data Max Nodes Read Continuation Point	False

6.5.32 Historical Access Structured Data Server Facet

Table 53 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 53 – 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 Time Instance	True
Historical Access	Historical Access Structured Data Update	True
Historical Access	Historical Access Structured Data Replace	True

6.5.33 Historical Data AtTime Server Facet

Table 54 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 54 – 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

6.5.34 Historical Access Modified Data Server Facet

Table 55 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 55 – 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

6.5.35 Historical Annotation Server Facet

Table 56 describes the details of the Historical Annotation *Server* Facet. This Facet defines support for the storage and retrieval of annotations for historical data.

Table 56 – 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

6.5.36 Historical Data Update Server Facet

Table 57 describes the details of the Historical Data Update *Server* Facet. This Facet includes Historical Data Update functionality.

Table 57 – 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

6.5.37 Historical Data Replace Server Facet

Table 57 Table 58 describes the details of the Historical Data Replace *Server* Facet. This Facet includes Historical Data Replace functionality.

Table 58 – Historical Data Replace Server Facet

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

6.5.38 Historical Data Insert Server Facet

Table 59 describes the details of the Historical Data Insert *Server* Facet. This Facet includes Historical Data Insert functionality.

Table 59 – 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

6.5.39 Historical Data Delete Server Facet

Table 60 describes the details of the Historical Data Delete *Server* Facet. This Facet includes Historical Data Delete functionality.

Table 60 – Historical Data Delete Server Facet

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

6.5.40 Base Historical Event Server Facet

Table 61 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 61 – 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

6.5.41 Historical Event Update Server Facet

Table 62 describes the details of the Historical *Event* Update *Server* Facet. This Facet includes Historical *Event* update access functionality.

Table 62 – Historical Event Update Server Facet

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

6.5.42 Historical Event Replace Server Facet

Table 62 describes the details of the Historical *Event* Replace *Server* Facet. This Facet includes Historical *Event* replace access functionality.

Table 63 – Historical Event Replace Server Facet

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

6.5.43 Historical Event Insert Server Facet

Table 64 describes the details of the Historical *Event* Insert *Server* Facet. This Facet includes Historical *Event* insert access functionality.

Table 64 – Historical Event Insert Server Facet

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

6.5.44 Historical Event Delete Server Facet

Table 65 describes the details of the Historical *Event* Delete *Server* Facet. This Facet includes Historical *Event* delete access functionality

Table 65 – Historical Event Delete Server Facet

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

6.5.45 Aggregate Subscription Server Facet

Table 66 describes the details of the Aggregate *Subscription Server* Facet. This Facet defines the handling of the aggregate filter when subscribing for *Attribute* values.

Table 66 – 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

6.5.46 Nano Embedded Device Server Profile

Table 67 describes the details of the Nano Embedded Device *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.

Table 67 – Nano Embedded Device Server Profile

Group	Conformance Unit / *Profile* Title	Optional
Profile	Core Server Facet	False
Profile	UA-TCP UA-SC UA Binary	False

6.5.47 Micro Embedded Device Server Profile

Table 68 describes the details of the Micro Embedded Device *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 must be exposed.

Table 68 – Micro Embedded Device Server Profile

Group	Conformance Unit / *Profile* Title	Optional
Profile	Embedded DataChange Subscription Server Facet	False
Profile	Nano Embedded Device Server Profile	False
Base Information	Base Info Custom Type System	False
Session Services	Session Minimum 2 Parallel	False

6.5.48 Embedded UA Server Profile

Table 69 describes the details of the Embedded UA *Server Profile*. This *Profile* is a *FullFeatured Profile* that is intended for devices with more than 50 MBs 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 Policy – Basic128Rsa15 Facet, 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.

Table 69 – Embedded UA Server Profile

Group	Conformance Unit / *Profile* Title	Optional
Profile	Micro Embedded Device Server Profile	False
Profile	SecurityPolicy – Basic128Rsa15	False
Profile	Standard DataChange Subscription Server Facet	False
Profile	User Token – X509 Certificate Server Facet	False
Base Information	Base Info Engineering Units	True
Base Information	Base Info Placeholder Modelling Rules	True
Base Information	Base Info Type System	False
Security	Security Default ApplicationInstanceCertificate	False

6.5.49 Standard UA Server Profile

Table 70 describes the details of the Standard UA *Server Profile*. This *Profile* is a *FullFeatured Profile* that defines a minimum set of functionality required for PC based OPC UA servers. Such a server must provide the base *AddressSpace* structure with type nodes, instance nodes and diagnostic information. The *Server* must provide connection establishment through the OPC UA TCP binary protocol with security and the creation of at least 50 parallel sessions. It includes view services like browsing and the attribute services for reading and writing of current values. In addition, the monitoring of data changes is included

with a minimum of 5 subscriptions for half of the required sessions (total 225) and a minimum of 500 monitored items for half of the subscriptions (total 56250).

Table 70 – Standard UA Server Profile

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

6.5.50 Core Client Facet

Table 71 describes the details of the Core *Client* Facet. This Facet defines the core functionality required for any *Client*. This Facet includes the core functions for Security and *Session* handling.

Table 71 – Core Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Profile	SecurityPolicy – Basic128Rsa15	False
Profile	SecurityPolicy – None	False
Profile	User Token – User Name Password Client Facet	False
Profile	User Token – X509 Certificate Client Facet	False
Security	Security Administration	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

6.5.51 Base Client Behaviour Facet

Table 72 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 72 – Base Client Behaviour Facet

Group	Conformance Unit / *Profile* Title	Optional
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

6.5.52 Discovery Client Facet

Table 73 describes the details of the *Discovery Client* Facet. This Facet defines the ability to discover *Servers* and their Endpoints.

Table 73 – 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

6.5.53 AddressSpace Lookup Client Facet

Table 74 describes the details of the *AddressSpace* Lookup *Client* Facet. This Facet defines the ability to navigate through the *AddressSpace*and includes basic AddressS*pace* concepts, view and browse functionality and simple attribute read functionality.

Table 74 – 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
Base Information	Base Info Client Basic	False
Base Information	Base Info Client Change Events	True
Base Information	Base Info Client GetMonitoredItems Method	True
Base Information	Base Info Client Progress Events	True
Base Information	Base Info Client System Status	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 TranslateBrowsePath	True

6.5.54 Entry-Level Support Client Facet

Table 75 describes the details of the Entry-Level Support *Client* Facet. This Facet defines the ability to interoperate with low-end *Servers*, e.g. *Servers* that support the Nano Embedded *Profile* (either by automatically adapting to the *Server* capabilities or through configuration). It implies respecting *Server* provided limits for *Session*, continuation points, *Subscription*, user authorization and locales.

Table 75 – Entry-Level SupportClient Facet

Group	Conformance Unit / *Profile* Title	Optional
Session Services	Client Entry-Level Support	False

6.5.55 Multi-Server Client Connection Facet

Table 76 describes the details of the Multi-*Server Client* Connection Facet. This Facet defines the ability for simultaneous access to multiple *Servers*.

Table 76 – Multi-Server Client Connection Facet

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

6.5.56 File Access Client Facet

Table 77 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 77 –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

6.5.57 Documentation – Client

Table 78 describes the details of the Documentation – *Client*. This Facet provides a list of user documentation that a *Client* application should provide.

Table 78 – 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

6.5.58 Attribute Read Client Facet

Table 79 describes the details of the *Attribute* Read *Client* Facet. This Facet defines the ability to read *Attribute* values of *Nodes*.

Table 79 – Attribute Read Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Attribute Services	Attribute Client Read Base	False
Attribute Services	Attribute Client Read Complex	True
Attribute Services	Attribute Client Read with proper Encoding	True

6.5.59 Attribute Write Client Facet

Table 80 describes the details of the *Attribute* Write *Client* Facet. This Facet defines the ability to write *Attribute* values of *Nodes*.

Table 80 – Attribute Write Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Attribute Services	Attribute Client Write Base	False
Attribute Services	Attribute Client Write Complex	True
Attribute Services	Attribute Client Write Quality & TimeStamp	True

6.5.60 DataChange Subscriber Client Facet

Table 81 describes the details of the DataChange Subscriber *Client* Facet. This Facet defines the ability to monitor *Attribute* values for data change.

Table 81 – DataChange Subscriber Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Monitored Item Services	Monitor Client by Index	False
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

6.5.61 DataAccess Client Facet

Table 82 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 analog and discrete data items and their quality of service.

Table 82 – DataAccess Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Address Space Model	Address Space Client Base	False
Address Space Model	Address Space Client Complex DataTypes	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 Basic	False
Data Access	Data Access Client Deadband	True
Data Access	Data Access Client SemanticChange	True

6.5.62 Event Subscriber Client Facet

Table 83 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 83 – 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

6.5.63 Notifier and Source Hierarchy Client Facet

Table 84 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 84 – 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

6.5.64 A & C Base ConditionClient Facet

Table 85 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 85 – 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 ConditionClasses Client	False
Alarms and Conditions	A & C Refresh Client	False

6.5.65 A & C Address Space Instance Client Facet

Table 86 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 86 – A & C Address Space Instance Client Facet

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

6.5.66 A & C Enable Client Facet

Table 87 describes the details of the A & C Enable *Client* Facet. This Facet defines the ability to enable and disable *Alarms*,

Table 87 – 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

6.5.67 A & C Alarm Client Facet

Table 88 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 88 – 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 Comment Client	True
Alarms and Conditions	A & C Confirm Client	True
Alarms and Conditions	A & C Discrete Client	False
Alarms and Conditions	A & C Off Normal Client	True
Alarms and Conditions	A & C Shelving Client	True
Alarms and Conditions	A & C Trip Client	True

6.5.68 A & C Exclusive Alarming Client Facet

Table 89 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 89 – 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

6.5.69 A & C Non-Exclusive Alarming Client Facet

Table 90 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 90 – 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

6.5.70 A & C Previous Instances Client Facet

Table 91 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 91 – 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

6.5.71 A & C Dialog Client Facet

Table 92 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 92 – 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

6.5.72 A & E Proxy Facet

Table 93 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 93 – 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 Off Normal Client	False
Alarms and Conditions	A & C Refresh Client	False
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
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

6.5.73 Method Client Facet

Table 94 describes the details of the *Method Client* Facet. This Facet defines the ability to call arbitrary *Method*s.

Table 94 – Method Client Facet

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

6.5.74 Auditing Client Facet

Table 95 describes the details of the Auditing *Client* Facet. This Facet defines the ability to monitor *AuditEvents*.

Table 95 – 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

6.5.75 Node Management Client Facet

Table 96 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 96 – 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

6.5.76 Advanced Type Programming Client Facet

Table 97 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 97 – 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

6.5.77 Diagnostic Client Facet

Table 98 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 98 – 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

6.5.78 Redundant Client Facet

Table 99 describes the details of the Redundant *Client* Facet. This Facet defines the ability to use the redundancy feature available for redundant *Clients*.

Table 99 – Redundant Client Facet

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

6.5.79 Redundancy Switch Client Facet

Table 100 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 100 – Redundancy Switch Client Facet

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

6.5.80 Historical Access Client Facet

Table 101 describes the details of the Historical Access *Client* Facet. This Facet defines the ability to read, process, and update historical data.

Table 101 – 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

6.5.81 Historical Annotation Client Facet

Table 102 describes the details of the Historical Annotation *Client* Facet. This Facet defines the ability to retrieve and write annotations for historical data.

Table 102 – 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

6.5.82 Historical Data AtTime Client Facet

Table 103 describes the details of the Historical Data AtTime *Client* Facet. This Facet defines the ability to access data at specific instances in time.

Table 103 – 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

6.5.83 Historical Aggregate Client Facet

Table 104 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 104 – 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

6.5.84 Historical Data Update Client Facet

Table 105 describes the details of the Historical Data Update *Client* Facet. This Facet defines the ability to update historical data.

Table 105 – 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

6.5.85 Historical Data Replace Client Facet

Table 105 describes the details of the Historical Data Replace *Client* Facet. This Facet defines the ability to replace historical data.

Table 106 – 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

6.5.86 Historical Data Insert Client Facet

Table 107 describes the details of the Historical Data Insert *Client* Facet. This Facet defines the ability to insert historical data.

Table 107 – 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

6.5.87 Historical Data Delete Client Facet

Table 108 describes the details of the Historical Data Delete *Client* Facet. This Facet defines the ability to delete historical data.

Table 108 – 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

6.5.88 Historical Access Client Server Timestamp Facet

Table 109 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 109 – Historical Access Client Server Timestamp Facet

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

6.5.89 Historical Access Modified Data Client Facet

Table 110 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 110 – 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

6.5.90 Structured Data AtTime Client Facet

Table 111 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 111 – 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

6.5.91 Historical Structured Data Access Client Facet

Table 112 describes the details of the Historical Structured Data Access *Client* Facet. This Facet defines the ability to read structured values for historical nodes.

Table 112 – 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

6.5.92 Historical Structured Data Modified Client Facet

Table 113 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 113 – 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

6.5.93 Historical Structured Data Delete Client Facet

Table 114 describes the details of the Historical Structured Data Delete *Client* Facet. This Facet defines the ability to remove structured historical data.

Table 114 – 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

6.5.94 Historical Structured Data Update Client Facet

Table 115 describes the details of the Historical Structure Data Update *Client* Facet. This Facet defines the ability to update structured historical data.

Table 115 – 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

6.5.95 Historical Structured Data Replace Client Facet

Table 115 describes the details of the Historical Structure Data Replace *Client* Facet. This Facet defines the ability to replace structured historical data.

Table 116 – Historical Structured Data Replace Client Facet

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

6.5.96 Historical Structured Data Insert Client Facet

Table 117 describes the details of the Historical Structured Data Insert *Client* Facet. This Facet defines the ability to insert structured historical data.

Table 117 – 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

6.5.97 Historical Events Client Facet

Table 118 describes the details of the Historical Events *Client* Facet. This Facet defines the ability to read Historical Events, including simple filtering.

Table 118 – 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

6.5.98 Historical Event Update Client Facet

Table 119 describes the details of the Historical *Event* Update *Client* Facet. This Facet defines the ability to update historical events.

Table 119 – 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

6.5.99 Historical Event Replace Client Facet

Table 119 describes the details of the Historical *Event* Replace *Client* Facet. This Facet defines the ability to replace historical events.

Table 120 – 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

6.5.100 Historical Event Delete Client Facet

Table 121 describes the details of the Historical *Event* Delete *Client* Facet. This Facet defines the ability to delete of Historical events.

Table 121 – 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

6.5.101 Historical Event Insert Client Facet

Table 122 describes the details of the Historical *Event* Insert *Client* Facet. This Facet defines the ability to insert historical events.

Table 122 – 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

6.5.102 Aggregate Subscriber Client Facet

Table 123 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 123 – Aggregate Subscriber Client Facet

Group	Conformance Unit / *Profile* Title	Optional
Aggregates	Aggregate Subscription – Client DeltaBounds	True
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 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 NumberOfTransition	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 StandardDevPopulation	True
Aggregates	Aggregate Subscription – Client StandardDevSample	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

Group	Conformance Unit / *Profile* Title	Optional
Subscription Services	Subscription Client Basic	False
Subscription Services	Subscription Client Modify	True
Subscription Services	Subscription Client Multiple	True
Subscription Services	Subscription Client Republish	True

6.5.103 User Token – Anonymous Facet

Table 124 describes the details of the User Token – Anonymous Facet. This Facet indicates that anonymous User Tokens are supported.

Table 124 – User Token – Anonymous Facet

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

6.5.104 User Token – User Name Password Server Facet

Table 125 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 Activate *Session Service*.

Table 125 – User Token – User Name Password Server Facet

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

6.5.105 User Token – X509 Certificate Server Facet

Table 126 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 126 – User Token – X509 Certificate Server Facet

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

6.5.106 User Token – Issued Token Server Facet

Table 127 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 127 – User Token – Issued Token Server Facet

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

6.5.107 User Token – Issued Token Windows Server Facet

Table 128 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 128 – User Token – Issued Token Windows Server Facet

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

6.5.108 User Token – User Name Password Client Facet

Table 129 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 129 – User Token – User Name Password Client Facet

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

6.5.109 User Token – X509 Certificate Client Facet

Table 130 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 130 – User Token – X509 Certificate Client Facet

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

6.5.110 User Token – Issued Token Client Facet

Table 131 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 131 – User Token – Issued Token Client Facet

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

6.5.111 User Token – Issued Token Windows Client Facet

Table 132 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 132 – User Token – Issued Token Windows Client Facet

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

6.5.112 UA-TCP UA-SC UA Binary

Table 133 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 133 – UA-TCP UA-SC UA Binary

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

6.5.113 SOAP-HTTP WS-SC UA XML

Table 134 describes the details of the SOAP-HTTP WS-SC UA XML. This transport Facet defines a combination of network protocol, security protocol and message encoding that provides maximum compatibility with enterprise class web service applications through the use of XML encoded SOAP messages. The performance of this transport profile will not be as good as the profiles with binary encoded messages. It requires support for SOAP 1.2, WS-Secure Conversation and the UA XML Encoding 1.0

Table 134 – SOAP-HTTP WS-SC UA XML

Group	Conformance Unit / *Profile* Title	Optional
Protocol and Encoding	Protocol Soap Xml WS Security	False

6.5.114 SOAP-HTTP WS-SC UA Binary

Table 135 describes the details of the SOAP-HTTP WS-SC UA Binary. This transport Facet defines a combination of network protocol, security protocol and message encoding that balances compatibility with enterprise class web service applications and performance through the use of SOAP message bodies that contain UA binary encoded messages. It requires support for SOAP 1.2, WS-Secure Conversation and the UA Binary Encoding 1.0.

Table 135 – SOAP-HTTP WS-SC UA Binary

Group	Conformance Unit / *Profile* Title	Optional
Protocol and Encoding	Protocol Soap Binary WS Security	False

6.5.115 SOAP-HTTP WS-SC UA XML-UA Binary

Table 136 describes the details of the SOAP-HTTP WS-SC UA XML-UA Binary. This transport Facet combines the SOAP-HTTP WS-SC UA Binary and SOAP-HTTP WS-SC UA XML Facets. It is used by *Servers* that allow the *Client* to choose whether messages are encoded with XML or Binary. It requires support for SOAP 1.2, WS-Secure Conversation and the UA Binary Encoding 1.0 and the UA XML Encoding 1.0.

Table 136 – SOAP-HTTP WS-SC UA XML-UA Binary

Group	Conformance Unit / *Profile* Title	Optional
Protocol and Encoding	Protocol Soap Binary WS Security	False
Protocol and Encoding	Protocol Soap Xml WS Security	False

6.5.116 HTTPS UA Binary

Table 137 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 137 – HTTPS UA Binary

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

6.5.117 HTTPS UA XML

Table 138 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 toolkits. This transport requires that one of the TransportSecurity Profiles for TLS be provided.

Table 138 – HTTPS UA XML

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

6.5.118 Security User Access Control Full

Table 139 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 139 – 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

6.5.119 Security User Access Control Base

Table 140 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 140 – 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

6.5.120 Security Time Synchronization

Table 141 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 must be supported.

Table 141 – 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

6.5.121 Best Practice – Audit Events

Table 142 describes the details of the Best Practice – Audit Events. Subscriptions for Audit Events shall be restricted to authorized personnel.

Table 142 – Best Practice – Audit Events

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

6.5.122 Best Practice – Alarm Handling

Table 143 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 143 – Best Practice – Alarm Handling

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

6.5.123 Best Practice – Random Numbers

Table 144 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 144 – Best Practice – Random Numbers

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

6.5.124 Best Practice – Timeouts

Table 145 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 145 – Best Practice – Timeouts

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

6.5.125 Best Practice – Administrative Access

Table 146 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 146 – Best Practice – Administrative Access

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

6.5.126 Best Practice – Strict Message Handling

Table 147 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 147 – Best Practice – Strict Message Handling

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

6.5.127 Best Practice – Audit Events Client

Table 148 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 148 – Best Practice – Audit Events Client

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

6.5.128 SecurityPolicy – None

Table 149 describes the details of the SecurityPolicy – None. This security Facet defines a SecurityPolicy used for configurations with the lowest security needs. This SecurityPolicy can affect the behaviour of the CreateSession and Activate *Session* services. It also results in a SecureChannel which has no Channel Security. By default this SecurityPolicy should be disabled if any other SecurityPolicies are available.

Table 149 – SecurityPolicy – None

Group	Conformance Unit / *Profile* Title	Optional
Security	Security None	False
Security	Security None CreateSession ActivateSession	False

6.5.129 SecurityPolicy – Basic128Rsa15

Table 150 describes the details of the SecurityPolicy – Basic128Rsa15. This security Facet defines a Security Policy for configurations with medium security. It requires a PKI infrastructure.

As computing power increases, SecurityPolicies 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. NIST recommends users of this SecurityPolicy should consider upgrading it for key lengths less than 2048 in 2010. NIST also recommends that this SecurityPolicy should be deprecated in 2012 for key lengths less than

2048. It is recommended that *Servers* and *Client* support all security profiles and developers provide the recommended profile as a default. It is up to an administrator to configure the actual exposed SecurityPolicies.

Table 150 – SecurityPolicy – Basic128Rsa15

Group	Conformance Unit / *Profile* Title	Optional
Security	Security Basic 128Rsa15	False
Security	Security Certificate Validation	False
Security	Security Encryption Required	False
Security	Security Level 1	False
Security	Security Signing Required	False

6.5.130 SecurityPolicy – Basic256

Table 151 describes the details of the SecurityPolicy – Basic256. This security Facet defines a Security Policy for configurations with medium to high security needs. It requires a PKI infrastructure.

As computing power increases, SecurityPolicies 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. NIST recommends users of this SecurityPolicy should consider upgrading it for key sizes less than 2048 in 2010. NIST also recommends that this SecurityPolicy should be deprecated in 2012 for key sizes less than 2048. It is recommended that *Servers* and *Client* support all security profiles and developers provide the recommended profile as a default. It is up to an administrator to configure the actual exposed SecurityPolicies.

Table 151 – SecurityPolicy – Basic256

Group	Conformance Unit / *Profile* Title	Optional
Security	Security Basic 256	False
Security	Security Certificate Validation	False
Security	Security Encryption Required	False
Security	Security Level 2	False
Security	Security Signing Required	False

6.5.131 SecurityPolicy – Basic256Sha256

Table 152 describes the details of the SecurityPolicy – Basic256Sha256. This security Facet defines a Security Policy for configurations with high security needs. It requires a PKI infrastructure.

As computing power increases, SecurityPolicies 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 *Client* support all security profiles and developers provide the recommended profile as a default. It is up to an administrator to configure the actual exposed SecurityPolicies.

Table 152 – SecurityPolicy – Basic256Sha256

Group	Conformance Unit / *Profile* Title	Optional
Security	Security Basic 256 Sha256	False
Security	Security Level 3	False

6.5.132 TransportSecurity – TLS 1.0

Table 153 describes the details of the TransportSecurity- TLS 1.0 Profile. This Facet defines a transport security for configurations with medium high security needs. It makes uses of TLS_RSA_WITH_RC4_128_SHA. This security profile is less secure than TLS 1.2.

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 already recommends users of this TransportSecurity should upgrade to TLS 1.2. This Policy is listed for systems that do not support TLS 1.1 or 1.2. It is recommended that *Servers* and *Client* 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 153 – TransportSecurity – TLS 1.0

Group	Conformance Unit / *Profile* Title	Optional
Security	Security Level 1	False
Security	Security TLS_RSA_WITH_RC4_128_SHA	False

6.5.133 TransportSecurity – TLS 1.1

Table 154 describes the details of the TransportSecurity- TLS 1.1. This Facet defines a transport security for configurations with medium high security needs. This security profile is less secure than TLS 1.2.

As computing power increases, security algorithms 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. NIST recommends users of this TransportSecurity should consider upgrading to TLS 1.2. This transport security is provided for systems that do not support TLS 1.2. It is recommended that *Servers* and *Client* support all security profiles and developers provide provided the recommended profile as a default. It is up to an administrator to configure the actual exposed TransportSecurity Profiles.

Table 154 – TransportSecurity – TLS 1.1

Group	Conformance Unit / *Profile* Title	Optional
Security	Security Level 2	False
Security	Security TLS 1.1	False

6.5.134 TransportSecurity – TLS 1.2

Table 155 describes the details of the SecurityPolicy – 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 *Client* 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 155 – TransportSecurity – TLS 1.2

Group	Conformance Unit / *Profile* Title	Optional
Security	Security Level 3	False
Security	Security TLS_RSA_WITH_AES_256_CBC_SHA256	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 deuxième édition annule et remplace la première édition parue en 2012. Cette édition constitue une révision technique.

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

a) Ajout d'un nombre important de nouvelles Facettes pour couvrir des additional functional areas of OPC UA. Plus particulièrement:
   – Facettes pour l'Accès à l'historique;

- Facettes pour les Agrégats;
- Facettes pour les HTTP
- Nouvelles facettes de sécurité
- Nouvelle facette de Jeton Utilisateur prenant en charge l'accès Anonyme
- Facettes Meilleures Pratiques,

b) Nouvelle Politique de Sécurité pour longueur de clé asymétrique > 2048

Le texte de cette norme est issu des documents suivants:

CDV	Rapport de vote
65E/378/CDV	65E/406/RVC

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

Cette publication a été rédigée selon les Directives ISO/IEC, Partie 2.

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 cette publication 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 à la publication recherchée. À cette date, l'édition sera

- reconduite,
- supprimée,
- remplacée par une édition révisée, ou
- amendée.

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 décrit les *Profils* de l'Architecture unifiée OPC (OPC UA). Les *Profils* du présent document permettent de dissocier les caractéristiques relatives aux essais des produits OPC UA, ainsi que la nature des essais (basés sur un outil ou en laboratoire). Ceci inclut les essais effectués par l'outil d'essai de conformité (CTT) OPC UA de la Fondation OPC (outil autonome pour les essais), ainsi que les essais réalisés par des laboratoires de certification indépendants de cette même fondation. On pourrait faire référence aux outils d'essai ou au laboratoire d'essai d'un autre organisme. L'élément important dans le cas présent est le concept qui oppose les essais basés sur un outil automatisé aux 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 d'une fonctionnalité à appliquer lors des essais de produits OPC UA en laboratoire ou en utilisant des outils automatisés. Contrairement aux catégories générales des Cas d'Essai, la définition des *Cas d'Essai* (*TestCases*) réels ne relève pas du domaine d'application du présent document.

La plupart des applications OPC UA sont conformes à de nombreux *Profils*, mais pas à la totalité des *Profils*.

2 Références normatives

Les documents suivants sont cités en référence de manière normative, en intégralité ou en partie, dans le présent document et sont indispensables pour son application. 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, *OPC unified architecture – Part 3: Address space model* (disponible en anglais seulement)

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: Correspondances*

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[1], *OPC unified architecture – Part 11: Historical access* (disponible en anglais seulement)

IEC 62541-13[1], *OPC unified architecture – Part 13: Aggregates* (disponible en anglais seulement)

3 Termes, définitions et conventions

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. Une vue d'ensemble des termes définis dans la présente Norme et leurs interactions peut être trouvée à la Figure 1.

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
unité de conformité
ConformanceUnit
ensemble spécifique de caractéristiques OPC UA pouvant être soumises à l'essai comme une entité unique

Note 1 à l'article: Une *Unité de Conformité* peut couvrir un groupe de services, des parties de services ou des modèles d'informations. Pour plus d'informations, voir l'Article 5.

3.1.3
groupe de conformité
ConformanceGroup
groupe de *ConformanceUnits* (unités de conformité) auquel un nom est attribué

Note 1 à l'article: Ce regroupement est destiné uniquement à faciliter l'organisation des *Unités de Conformité*. Les *Groupes de Conformité* typiques incluent les groupes propres à chacun des ensembles de service de l'OPC UA et à chacune des normes du Modèle d'informations.

3.1.4
facette
Facet
profil dédié à une caractéristique spécifique qu'un *Serveur* ou *Client* est susceptible d'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* complet**
FullFeatured *Profile*
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.

[1] A publier.

3.1.6
catégorie de profil
ProfileCategory
organise les *Profils* en classes d'application, 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
cas d'essai
TestCase
description technique d'un ensemble d'étapes nécessaires pour soumettre à essai une fonction ou un modèle d'informations particulier

Note 1 à l'article: Les *Cas d'Essai* fournissent des détails suffisants pour permettre à un développeur de les mettre en œuvre dans un code. Les *Cas d'Essai* fournissent également un résumé détaillé du ou des résultats prévus d'exécution du code mis en œuvre, ainsi que toute(s) condition(s) préalable(s) devant être établie(s) avant de pouvoir exécuter le *Cas d'Essai*.

3.1.8
laboratoire d'essai
TestLab
installation conçue pour fournir des services d'essai

Note 1 à l'article: Ces services incluent, sans toutefois s'y limiter, le personnel qui effectue directement les essais, les essais automatisés et un processus reproductible formel. La Fondation OPC a fourni une norme détaillée décrivant les Laboratoires d'Essai OPC UA ainsi que les essais à organiser (se reporter à Compliance Part 8 UA *Server*, Compliance Part 9 UA *Client*).

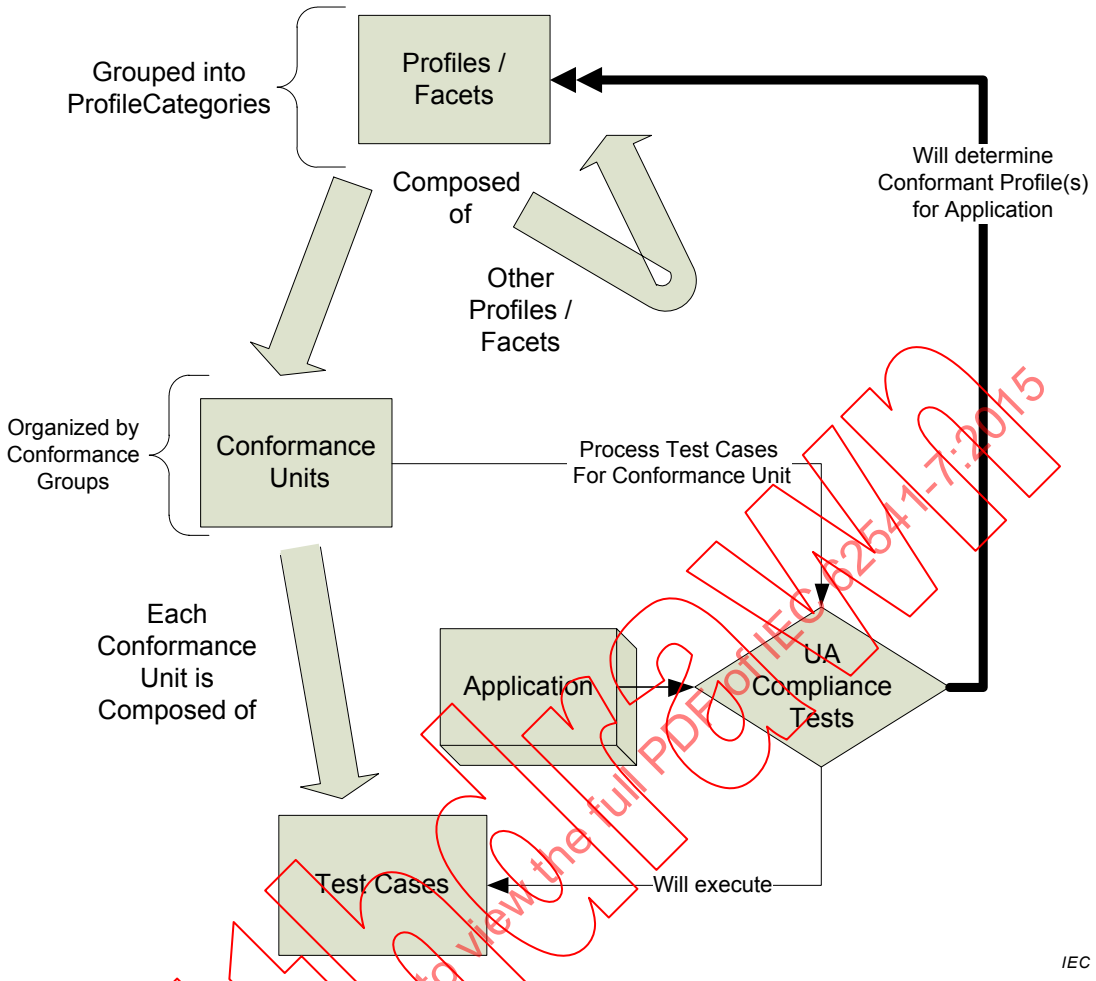
3.2 Abréviations

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 norme en plusieurs parties de l'architecture Unifiée OPC décrit un certain nombre de *Services* et divers modèles d'informations. Ces *Services* et modèles d'informations peuvent être désignés comme les caractéristiques d'un *Serveur* ou d'un *Client*. Il est nécessaire que les *Serveurs* et *Clients* soient également capables de décrire les caractéristiques qu'ils prennent en charge et souhaitent voir certifiées. Le présent document fournit un regroupement de ces caractéristiques. Les caractéristiques individuelles sont regroupées en *Unités de Conformité* qui sont par ailleurs regroupées en *Profils*. La Figure 1 présente un aperçu général des interactions entre les *Profils*, les *Unités de Conformité* et les *Cas d'Essai*. Les grandes flèches désignent les composants utilisés pour composer le parent. Par exemple, un *Profil* est constitué à partir des *Profils* et des *Unités de Conformité*. La figure illustre également une caractéristique de l'Outil d'Essai de Conformité (CTT, Compliance Test Tool) OPC UA, en ce sens qu'elle permet de vérifier par essai si un *Profil* invoqué satisfait à toutes les *Unités de Conformité*. L'outil permet également de vérifier par essai toutes les autres *Unités de Conformité* et de consigner dans un rapport tous les autres *Profils* qui satisfont aux essais de conformité. Les *Cas d'Essai* individuels sont définis dans des documents séparés; se reporter à Compliance Part 8 UA Server et Compliance Part 9 UA Client. Les *Cas d'Essai* sont associés en retour aux *Unités de Conformité* appropriées définies dans la présente Norme. Cette relation est également affichée par l'outil d'essai de conformité OPC UA.



Légende

Anglais	Français
Grouped into ProfileCategories	Regroupés dans des Catégories de Profil
Profiles / Facets	Profils / Facettes
Composed of	Constitués de
Other Profiles / Facets	Autres Profils / Facettes
Will determine Conformant Profile(s) for Application	Déterminent le(s) Profil(s) Conforme(s) pour Application
Organized by Conformance Groups	Organisés par Groupes de Conformité
Conformance Units	Unités de Conformité
Process Test Cases For Conformance Unit	Cas d'Essai de Processus pour Unité de Conformité
Each Conformance Unit is Composed of	Chaque Unité de Conformité est Constituée de
UA Compliance Tests	Essais de Conformité UA
Test Cases	Cas d'Essai
Will execute	Exécute

Figure 1 – Profil – Unité de Conformité – Cas d'Essai

4.2 Unité de Conformité

Chaque *Unité de Conformité* représente un ensemble spécifique de caractéristiques (par exemple, un groupe de services, des parties de services ou des modèles d'informations) pouvant être soumises à l'essai en tant qu'entité unique. Les *Unités de Conformité* sont les

blocs fonctionnels d'un *Profil*. Chaque *Unité de Conformité* peut également être utilisée comme catégorie d'essai. Il existe, pour chaque *Unité de Conformité,* un certain nombre de Cas d'Essai qui soumettent à l'essai la fonctionnalité décrite par l'*Unité de Conformité.* La description d'une *Unité de Conformité* est destinée à fournir des informations suffisantes pour illustrer la fonctionnalité requise, mais, dans de nombreux cas, pour avoir une compréhension complète de l'*Unité de Conformité* le lecteur peut avoir besoin d'examiner également la partie appropriée de l'IEC 62541. Des informations supplémentaires concernant les essais d'une *Unité de Conformité* 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 *Unités de Conformité*.

4.3 Profils

Un *Profil* est une agrégation nommée d'*Unités de Conformité* et autres *Profils*. Pour prendre en charge un *Profil*, il faut qu'une application prenne en charge les *Unités de Conformité* et tous les *Profils* agrégés. La définition des *Profils* est une activité continue, en ce sens qu'il est prévu d'ajouter ultérieurement de nouveaux *Profils*.

Une application OPC UA prend généralement en charge des *Profils* multiples.

Ces *Profils* multiples peuvent comporter la même *Unité de Conformité*.

Soumettre à l'essai un *Profil* consiste à soumettre à l'essai les *Unités de Conformité* individuelles qui comportent le *Profil*.

La désignation des *Profils* s'effectue sur la base des conventions d'affectation des noms (voir 6.3 pour des informations détaillées).

4.4 Catégories de Profils

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

Le Tableau 1 contient la liste des *Catégories de Profil* actuellement définies.

Tableau 1 – ProfileCategories

Catégorie	Description
Client	Les *Profils* de cette catégorie spécifient les fonctions d'un Client OPC UA. L'URI de ce type de *Profils* peut faire partie intégrante d'un *Certificat* de Logiciel transmis dans la requête ActivateSession (CréerSession).
Sécurité	Les *Profils* de cette catégorie spécifient les fonctions relatives à la Sécurité. Les politiques de sécurité font partie de cette catégorie. Il faut que l'URI de ces politiques de sécurité fasse partie intégrante d'une Description du point d'extrémité transmise par le service GetEndpoint (Obtenir Point d'extrémité). Les *Profils* de cette catégorie s'appliquent aux *Serveurs* et aux *Clients*.
Serveur	Les *Profils* de cette catégorie spécifient des fonctions pour un *Serveur* OPC UA. L'URI de ce type de *Profils* peut faire partie intégrante d'un *Certificat de logiciel* transmis avec la réponse du service CreateSession et présenté dans les capacités du *Serveur*.
Transport	Les *Profils* de cette catégorie spécifient des correspondances de protocoles spécifiques. Il faut que l'URI de ce type de *Profils* fasse partie intégrante d'une Description du point d'extrémité. Ces *Profils* s'appliquent aux *Serveurs* et aux *Clients*.

5 Unités de Conformité

5.1 Vue d'ensemble

Une *Unité de Conformité* représente une entité pouvant être soumise individuellement à l'essai. Pour plus de clarté, la longue liste des *Unités de Conformité* est regroupée en *Groupes de Conformité*. Ces groupes reflètent les *Jeux de Services* définis dans l'IEC 62541-4 et les modèles d'informations OPC UA. Le Tableau 2 fournit la liste des *Groupes de Conformité*. Ces groupes et les *Unités de Conformité* qu'ils décrivent sont détaillés dans les Paragraphes de l'Article 5 commençant au Paragraphe 5.2. Les *Groupes de Conformité* n'ont aucune influence sur les essais; ils sont utilisés uniquement pour des raisons d'organisation, c'est-à-dire pour simplifier la lisibilité du présent document.

Tableau 2 – ConformanceGroups

Groupe	Description
Modèle de l'Espace d'adresses	Définit les *Unités de Conformité* pour diverses caractéristiques de l'*Espace d'adresses* OPC UA.
Agrégats	Ensemble des *Unités de Conformité* relatives aux Agrégats, y compris les *Unités de Conformité* individuelles pour chaque Agrégat pris en charge comme décrit dans l'IEC 62541-13.
Alarmes et Conditions	Ensemble des *Unités de Conformité* associées au modèle d'informations OPC UA pour les *Conditions*, *Conditions* acceptables, confirmations et *Alarmes* comme spécifié dans l'IEC 62541-9.
Services Attribut	Inclut les *Unités de Conformité* permettant de lire ou d'écrire les valeurs d'*Attribut* actuelles ou historiques.
Audit	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 comme défini dans l'IEC 62541-5.
Accès aux données	*Unités de Conformité* spécifiques aux *Clients* et aux *Serveurs* qui traitent de la représentation et de l'utilisation des données d'automatisation, comme spécifié dans l'IEC 62541-8.
Services Découverte	*Unités de Conformité* axées sur la *Découverte* des Points d'extrémité du *Serveur*.
Accès à l'historique	Accès aux données archivées des valeurs d'*Attribut* de nœud ou Évènements.
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 *Unités de Conformité* qui couvrent des sujets divers, tels que les comportements recommandés, la documentation, etc. Typiquement ces *Unités de Conformité* ne correspondent à aucun des autres groupes.
Services Eléments Surveillés	Les *Clients* définissent des *MonitoredItems* pour s'abonner aux données et aux Événements. Chaque *MonitoredItem* identifie l'élément à surveiller et l'*Abonnement* (*Subscription*) à utiliser pour transmettre des *Notifications*.
Services Gestion des Nœuds	Groupe les *Unités de Conformité* pour le Jeu de *Services* afin d'ajouter et de supprimer des *Nœuds* et des *Références* de l'*Espace d'adresses* OPC UA.
Protocole et Codage	Couvre toutes les combinaisons de transport et de codage spécifiées dans l'IEC 62541-6.
Services de Consultation	Une Consultation peut être utilisée pour assurer un filtrage avancé et renvoyer un sous-ensemble de données.

Groupe	Description
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	*Unités de Conformité* relatives à la sécurité pouvant 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 signifier les *Notifications* au *Client*.
Services Vue	Les *Clients* utilisent le Jeu de Services Vue pour naviguer dans l'*Espace d'adresses* OPC UA ou dans une Vue (un sous-ensemble) de l'*Espace d'adresses* OPC UA.

5.2 Services

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

Une *Unité de Conformité* simple peut référencer plusieurs *Services* (par exemple, CreateSession (Créer Session), ActivateSession (Activer Session) et CloseSession (Fermer Session)), mais peut également faire référence aux aspects individuels des *Services* (par exemple, l'utilisation de ActivateSession pour masquer un nouvel utilisateur).

Chaque tableau comporte une liste de la *Catégorie de profil* à laquelle appartient une *Unité de Conformité*, le titre et la description de l'*Unité de Conformité,* ainsi qu'une colonne qui indique si l'*Unité de Conformité* est issue d'une autre *Unité de Conformité*. Ce type d'*Unité de Conformité* inclut tous les essais correspondant au parent auxquels s'ajoute(nt) un ou plusieurs Cas d'Essais supplémentaires. Ces Cas d'Essai peuvent uniquement limiter davantage les Cas d'Essai existants. Un exemple peut être un cas d'essai consistant à vérifier le nombre de connexions, où le Cas d'Essai du parent nécessite au moins une connexion et l'*Unité de Conformité* déduite nécessite un Cas d'Essai pour au moins cinq connexions.

Le Jeu de Services *Découverte* comprend plusieurs *Unités de Conformité* (voir Tableau 3). Tous les *Serveurs* fournissent certains aspects de cette fonctionnalité; se reporter aux *Profils* catégorisés comme *Profils de Serveur* pour des informations détaillées. Les *Clients* peuvent prendre en charge certains aspects de cette fonctionnalité; se reporter aux *Profils* catégorisés comme *Profils de Client* pour des informations détaillées.

Tableau 3 – Services Découverte

Catégorie	Titre	Description	Dérivation
Serveur	Discovery Get Endpoints (Obtenir les points d'extrémité)	Prend en charge le *Service* GetEndpoints afin d'obtenir tous les points d'extrémité du *Serveur*. Ceci inclut le filtrage basé sur les *Profils*.	
Serveur	Discovery Find Servers Self (Trouver les Serveurs à usage individuel)	Prend en charge le *Service* FindServers uniquement pour un usage individuel.	
Serveur	Discovery Register (Découverte de Registre)	Appelle le *Service* RegisterServer pour s'enregistrer (*Serveur* OPC UA) auprès d'un *Service Découverte* externe via un canal sécurisé par un SecurityMode (Mode de sécurité) autre que ''None'' ("Aucun").	
Serveur	Discovery Configuration (Découverte de Configuration)	Permet la configuration de l'URL du *Serveur de Découverte* où le *Serveur* s'enregistre lui-même. Permet la désactivation complète de l'enregistrement avec un *Serveur de Découverte*.	
Client	Discovery Client Find Servers Basic (Trouver les Serveurs de base du Client)	Utilise le *Service* FindServers pour obtenir tous les *Serveurs* installés sur une plate-forme donnée.	
Client	Discovery Client Find Servers with URI (Trouver les serveurs ayant un URI)	Utilise le *Service* FindServers pour obtenir les URL des URI des *Serveurs* spécifiques.	
Client	Discovery Client Find Servers Dynamic (Trouver les serveurs dynamiques client)	Détecte les nouveaux *Serveurs* après un appel initial du *Service* FindServers.	
Client	Discovery Client Get Endpoints Basic (Obtenir les points d'extrémité de base client)	Utilise le *Service* GetEndpoints pour obtenir tous les Points d'extrémité d'un URI de *Serveur* donné.	
Client	Discovery Client Get Endpoints Dynamic (Obtenir les points d'extrémité dynamiques client)	Détecte les changements opérés sur les Points d'extrémité après un appel initial du *Service* GetEndpoints.	
Client	Discovery Client Configure Endpoint (Configuration du Point d'extrémité Client)	Permet la spécification d'un Point d'extrémité sans passer par le Jeu de *Services Découverte*.	

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

Tableau 4 – Services Session

Catégorie	Titre	Description	Dérivation
Serveur	Session General Service Behaviour (Service Général - Comportement)	Met en œuvre le *Service* de base «comportement». Ceci inclut notamment: – la vérification du jeton d'authentification – la transmission de la requestHandle (requête Traitement) dans les réponses – la transmission des informations de diagnostic disponibles tel que demandée, avec le paramètre 'returnDiagnostics' (transmission de diagnostic) – le respect du message timeoutHint	
Serveur	Session de 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. Noter que pour les services CreateSession et ActivateSession, si le SecurityMode = None, alors: 1) Le *Certificat* d'application et Nonce sont facultatifs. 2) Les signatures sont nulles/vides. Ces détails sont décrits dans l'IEC 62541-4.	
Server	Session Modification Utilisateur	Prend en charge l'utilisation d'ActivateSession pour modifier l'utilisateur de la *Session*.	
Serveur	Session Cancel (Session Annuler)	Prend en charge le *Service* Cancel pour annuler les requêtes en suspens.	
Serveur	Session Minimum 1	Prend en charge la *Session* minimum 1 (total).	
Serveur	Session Minimum 2 Parallel (2 Sessions Minimum en Parallèle)	Prend en charge au minimum 2 sessions parallèles (total pour tous les *Clients*).	
Serveur	Session Minimum 50 Parallel (50 Sessions Minimum en Parallèle)	Prend en charge au minimum 50 sessions parallèles (total pour tous les *Clients*).	
Client	Session Client General Service Behaviour (service général Comportement client)	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 requête des informations de diagnostic avec le paramètre 'returnDiagnostics' – l'évaluation du serviceResult (résultat de service) et des résultats d'exploitation	

Catégorie	Titre	Description	Dérivation
Client	Session Client Base (Client de Base)	Utiliser 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. Noter que 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 Multiple Connections (Connexions multiples Client)	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. Une limite à cause de la mémoire est autorisée, mais pas une limite à cause de la capacité logicielle.	
Client	Session Client Renew NodeIds (Renouveler les identificateurs de nœuds Client)	Cette *Unité de Conformité* s'applique aux Clients qui autorisent les NodeIds conservés. Vérifier que le Tableau de Namespace (Espace de nom) 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 changements sont intervenus, il faut que le *Client* recalcule les indices de l'espace de nom des NodeIds respectifs.	
Client	Session Client Impersonate (Masquer Client)	Utilise ActivateSession pour changer l'utilisateur de la *Session* (masquage).	
Client	Session Client KeepAlive (Maintenir Client actif)	Formuler des requêtes périodiques de maintien de la *Session* active.	
Client	Session Client Detect Shutdown (Détecter un arrêt Client)	Lit ou surveille la *Variable* ServerStatus/State afin de reconnaître un arrêt potentiel du *Serveur* et de «nettoyer» les ressources.	
Client	Session Client Cancel (Annuler Client)	Utilise le *Service* Cancel pour annuler les requêtes en suspens.	
Client	Session Client Auto Reconnect (Reconnexion automatique Client)	Reconnexion automatique du Client y compris: – ActivateSession avec nouveau SecureChannel si le SecureChannel existant n'est plus valide, mais si la *Session* est toujours valide – la création d'une nouvelle *Session* uniquement si la *Session* n'est plus valide	

Catégorie	Titre	Description	Dérivation
Client	Client Entry-Level Support (prise en charge au niveau de l'entrée)	Le *Client* est capable d'interagir avec les *Serveurs* avec le plus faible niveau de fonctionnalité. Cela comprend la capacité de fonctionner avec une seule *Session*, des connaissances préalables des Types OPC UA (le *Serveur* ne peut pas les présenter dans l'*Espace d'adresses*), et la capacité à utiliser Read (Lecture) en fonction de *Subscriptions* (Abonnements) pour la surveillance. Il peut y avoir des restrictions supplémentaires imposées par le *Serveur* via les capacités du *Serveur*.	

Le Jeu de *Services Gestion* des *Nœuds* comprend plusieurs *Unités de Conformité* (voir Tableau 5). Les *Serveurs* peuvent fournir certains aspects de cette fonctionnalité, se reporter aux *Profils* catégorisés comme *Profils de Serveur* pour des informations détaillées. Les *Clients* peuvent prendre en charge certains aspects de cette fonctionnalité, se reporter aux *Profils* catégorisés comme *Profils de Client* pour des informations détaillées.

Tableau 5 – Services Gestion des Nœuds

Catégorie	Titre	Description	Dérivation
Serveur	Node Management Add Node (Ajouter des Nœuds)	Prend en charge le *Service* AddNodes pour ajouter un ou plusieurs *Nœuds* dans l'*Espace d'adresses* OPC UA.	
Serveur	Node Management Delete Node (Supprimer Nœud)	Prend en charge le *Service* DeleteNodes pour supprimer un ou plusieurs *Nœuds* de l'*Espace d'adresses* OPC UA.	
Serveur	Node Management Add Ref (Ajouter Référence)	Prend en charge le *Service* AddReferences pour ajouter une ou plusieurs *Références* à un ou plusieurs *Nœuds* dans l'*Espace d'adresses* OPC UA.	
Serveur	Node Management Delete Ref (Supprimer Référence)	Prend en charge le *Service* DeleteReferences pour supprimer une ou plusieurs *Références* d'un *Nœud* dans l'*Espace d'adresses* OPC UA.	
Client	Node Management 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'*Espace d'adresses* OPC UA du *Serveur*.	

Le Jeu de *Services* Vue comprend plusieurs *Unités de Conformité* (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 des informations détaillées.

Tableau 6 – Services Vue

Catégorie	Titre	Description	Dérivation
Serveur	Vue Basic (Vue de base)	Prend en charge le Jeu de *Services* Vue (Browse – Navigation, BrowseNext – Navigation suivante).	
Serveur	Vue TranslateBrowsePath (Traduire Chemin de navigation)	Prend en charge le *Service* TranslateBrowsePathsToNodeIds (Traduire Chemins de navigation en Identificateurs de nœuds).	
Serveur	Vue RegisterNodes (Enregistrement des nœuds)	Prend en charge les *Services* RegisterNodes et UnregisterNodes (Désinscription des nœuds) comme méthode d'optimisation de l'accès aux *Nœuds* à utilisation répétitive dans l'*Espace d'adresses* OPC UA du *Serveur*.	
Serveur	Vue Minimum Continuation Point 01 (01 Point de continuation minimum)	Prend en charge au minimum 1 point de continuation par *Session*.	
Serveur	Vue Minimum Continuation Point 05 (05 Points de continuation minimum)	Prend en charge au minimum 5 points de continuation par *Session*. Ce nombre est à prendre en charge pendant au moins la moitié des sessions requises minimales.	
Client	Vue Client Basic Browse (Navigation de base Client)	Utilise les *Services* Browse et BrowseNext pour naviguer dans l'*Espace d'adresses* OPC UA du *Serveur*. Utiliser referenceTypeId (identificateur de type de référence) et le nodeClassMask (Masque de classe de nœud) pour spécifier les *Références* nécessaires.	
Client	Vue Client Basic ResultSet Filtering (Filtrage de base de l'ensemble de résultats Client)	Utilise le paramètre ResultMask (Masque de résultat) 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 (chemin de navigation) sont connus. Effectuer des opérations d'ensemble plutôt que des appels multiples lorsque cela est possible.	
Client	Vue Client RegisterNodes (Enregistrement des nœuds)	Utilise le *Service* RegisterNodes pour optimiser l'accès aux *Nœuds* à utilisation répétitive. Utiliser le Service UnregisterNodes lorsque les *Nœuds* ne sont plus utilisés.	

Le Jeu de *Services Attribut* comprend plusieurs *Unités de Conformité* (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	Titre	Description	Dérivation
Serveur	Attribut Read (Lecture)	Prend en charge le *Service* Read afin de déchiffrer un ou plusieurs *Attributs* d'un ou plusieurs *Nœuds*. Ceci inclut la prise en charge du paramètre IndexRange (Intervalle d'indice) afin de déchiffrer un seul élément ou un ensemble d'éléments lorsque la valeur *Attribut* constitue une matrice.	
Serveur	Attribut Read Complex (Lecture Complexe)	Prend en charge la lecture et le codage de Valeurs avec types de données structurées.	
Serveur	Attribut Write Values (Ecrire des Valeurs)	Prend en charge l'écriture de valeurs à un ou plusieurs *Attributs* d'un ou plusieurs *Nœuds*.	
Serveur	Attribut Write Complex (Ecriture Complexe)	Prend en charge l'écriture et le codage de Valeurs avec types de données structurées.	
Serveur	Attribut Write StatusCode & TimeStamp (Ecrire Code de Statut & Horodatage)	Prend en charge l'écriture du StatusCode (Code de Statut) et des Timestamps (Horodatages) associés à la Valeur.	
Serveur	Attribut Write Index (Ecrire Indice)	Prend en charge le paramètre IndexRange pour écrire un seul élément ou un ensemble d'éléments lorsque la valeur *Attribut* constitue une matrice.	
Serveur	Attribut Alternate Encoding (Codage alternatif)	Prend en charge le codage alternatif des données lors de la lecture des *Attributs* de la Valeur. Par défaut, il faut que chaque *Serveur* prenne en charge le codage des données du *Profil* de Pile réellement utilisé (c'est-à-dire binaire avec le Codage Binaire UA et XML avec le Codage XML). Cette *Unité de Conformité* – lorsqu'elle est prise en charge – spécifie également la prise en charge de l'autre codage des données.	
Server	Attribute Historical Read (Lecture Historique)	Prend en charge le *Service* HistoryRead. Les détails des aspects utilisés par ce service sont énumérés dans des *Unités de Conformité* supplémentaires, mais au moins une des unités suivantes doit être prise en charge: ReadRaw, ReadProcessed, ReadModified, ReadAtTime ou ReadEvents.	
Server	Attribute Historical Update (Mise à jour Historique)	Prend en charge le *Service* Mise à jour Historique. Les détails des caractéristiques prises en charge de ce service sont décrits par des *Unités de Conformité* supplémentaires, mais au moins une des unités suivantes doit être prise en charge: InsertData, InsertEvents, ReplaceData, ReplaceEvents, UpdateData, UpdateEvents, DeleteData, DeleteEvents ou DeleteAtTime.	

Catégorie	Titre	Description	Dérivation
Client	Attribut Client Read Base (Lecture Base)	Utiliser le *Service* Read pour lire un ou plusieurs *Attributs* de un ou plusieurs *Nœuds*. Ceci inclut l'utilisation d'un IndexRange pour sélectionner un seul élément ou un ensemble d'éléments lorsque la valeur *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 Read with proper Encoding (Lecture avec codage approprié)	Cette *Unité de Conformité* fait référence à la capacité d'un *Serveur* à prendre en charge plusieurs codages de données pour les valeurs d'*Attribut*. Les *Clients* peuvent découvrir les codages disponibles et peuvent en choisir un de manière explicite lorsqu'ils appellent le *Service* Read.	
Client	Attribut Client Read Complex (Lecture Complexe)	Lecture et codage de Valeurs avec types de données structurées.	
Client	Attribut Client Write Base (Ecriture Base)	Utiliser le *Service* Write pour écrire des valeurs à un ou plusieurs *Attributs* d'un ou plusieurs *nœuds*. Ceci inclut l'utilisation d'un IndexRange afin de sélectionner un seul élément ou un ensemble d'éléments lorsque la valeur *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 Write Complex (Ecriture Complexe)	Écriture et codage de Valeurs avec types de données structurées.	
Client	Attribut Client Write Quality & TimeStamp (Ecriture Qualité & Horodatage)	Utiliser le *Service* Write pour écrire également le StatusCode et/ou les Timestamps associés à une Valeur.	
Client	Attribute Client Historical Read (Lecture Historique Client)	Le *Client* utilise le service HistoryRead. Les détails des aspects utilisés par ce service sont fournis par des *Unités de Conformité* supplémentaires, mais au moins une ou plusieurs des unités suivantes sont utilisées: ReadRaw, ReadAtTime, ReadProcessed, ReadModified ou ReadEvents.	
Client	Attribute Client Historical Updates (Mises à jour Historique Client)	Le *Client* utilise le service HistoryUpdate. Les détails de cette utilisation sont fournis par des *Unités de Conformité* supplémentaires, mais au moins une ou plusieurs des unités suivantes doivent être fournies: InsertData, InsertEvent, ReplaceData, ReplaceEvent, UpdateData, UpdateEvents, DeleteData ou DeleteEvents ou DeleteAtTime.	

Le Jeu de *Services Méthode* est constitué d'*Unités de Conformité* (voir Tableau 8). Les *Unités de Conformité* 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 des informations détaillées. Les *Clients* peuvent prendre en charge certains aspects de cette fonctionnalité, se reporter aux *Profils* catégorisés comme *Profils de Client* pour des informations détaillées.

Tableau 8 – Services Méthode

Catégorie	Titre	Description	Dérivation
Serveur	Méthode Call (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 Call (Appel Client)	Utiliser le *Service* Appel pour appeler une ou plusieurs Méthodes.	

Le Jeu de *Services Eléments Surveillés* (*MonitoredItem)* comprend plusieurs *Unités de Conformité* (voir Tableau 9). Les *Serveurs* peuvent fournir certains aspects de cette fonctionnalité; se reporter aux *Profils* catégorisés comme *Profils de Serveur* pour des informations détaillées. Les *Clients* peuvent prendre en charge certains aspects de cette fonctionnalité, se reporter aux *Profils* catégorisés comme *Profils de Client* pour des informations détaillées.

Tableau 9 – Services Eléments Surveillés

Catégorie	Titre	Description	Dérivation
Serveur	Contrôle de base	Prend en charge les *Services Eléments Surveillés* suivants: CreateMonitoredItems (Créer Éléments Surveillés), ModifyMonitoredItems (Modifier Éléments Surveillés), DeleteMonitoredItems (Supprimer Éléments Surveillés), SetMonitoringMode (Définir Mode de Surveillance).	
Serveur	Contrôle Value Change (Modification de valeur)	Prend en charge la création *d'Éléments Surveillés* pour les modifications de valeur d'*Attributs*. Ceci inclut la prise en charge d'IndexRange afin de sélectionner un seul élément ou un ensemble d'éléments lorsque la valeur *Attribut* constitue une matrice.	
Server	Filtre de Bande Morte d'Éléments Surveillés	Prend en charge un Filtre de Bande Morte absolu comme DataChangeFilter pour les types de données numériques.	
Server	Contrôle Filtre Aggrégat	Prend en charge les filtres Aggrégat pour les *MonitoredItems*. Le résultat de ces *Unités de Conformité* inclut une liste d'Aggrégats qui sont pris en charge en tant que partie du *Certificat de Profil.*	
Serveur	Contrôle Alternate Encoding (codage alternatif)	Prend en charge le codage alternatif lors de la surveillance des *Attributs* de la valeur. Par défaut, il faut que chaque *Serveur* prenne en charge le codage du *Profil* de Piles réellement utilisé (c'est-à-dire binaire avec le Codage Binaire UA et XML avec le Codage XML). Cette *Unité de Conformité* – lorsqu'elle est prise en charge – spécifie également la prise en charge de l'autre codage des données.	

Catégorie	Titre	Description	Dérivation
Serveur	Monitor Items 2 (Contrôle 2 éléments)	Prend en charge au moins 2 *Éléments Surveillés* par *Abonnement*	
Serveur	Monitor Items 10 (Contrôle 10 éléments)	Prend en charge au moins 10 *Éléments Surveillés* par *Abonnement*	
Serveur	Monitor Items 100 (Contrôle 100 éléments)	Prend en charge au moins 100 *Éléments Surveillés* par *Abonnement*. Il faut que ce nombre soit pris en charge pour au moins la moitié des Abonnements requis, pour la moitié des Sessions requises.	
Serveur	Monitor Items 500 (Contrôle 500 éléments)	Prend en charge au moins 500 *Éléments Surveillés* par *Abonnement* Il faut que ce nombre soit pris en charge pour au moins la moitié des Abonnements requis, pour la moitié des Sessions requises.	
Serveur	Contrôle QueueSize_1 (Taille de la File d'attente_1)	Cette *Unité de Conformité* ne requiert aucune mise en file d'attente en cas de plusieurs modifications de valeurs au cours d'une période d'édition, c'est-à-dire que la toute dernière modification sera transmise dans la *Notification*.	
Serveur	Contrôle MinQueueSize_02 (Au minimum 2 Files d'attente)	Prend en charge au moins 2 entrées de file d'attente pour les *Éléments Surveillés*. Les *Serveurs* adaptent souvent la taille de la file d'attente au nombre des *Éléments Surveillés* actuels. Toutefois, il est prévu que les *Serveurs* prennent en charge cette taille de file d'attente minimale pour au moins un tiers des *Éléments Surveillés* pris en charge.	
Serveur	Contrôle MinQueueSize_05 (Au minimum 5 Files d'attente)	Prend en charge au moins 5 entrées de file d'attente pour les *Éléments Surveillés*. Les *Serveurs* adaptent souvent la taille de la file d'attente au nombre des *Éléments Surveillés* actuels. Toutefois, il est prévu que les *Serveurs* prennent en charge cette taille de file d'attente minimale pour au moins un tiers des *Éléments Surveillés* pris en charge.	
Serveur	Contrôle QueueSize_ServerMax (La file d'attente c'est le max du Serveur)	Cette *Unité de Conformité* s'applique aux Événements. Lorsque le Client effectue une requête queuesize=MAXUInt32, il faut que le *Serveur* retourne 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 Triggering (Déclenchement)	Prend en charge le *Service* SetTriggering (Définir le déclenchement) pour créer et/ou supprimer les liens actifs d'un élément déclencheur.	

Catégorie	Titre	Description	Dérivation
Serveur	Contrôle Events (Événements)	Prend en charge la création des *Éléments Surveillés* pour un "*Attribut EventNotifier*" à des fins de *Notification d'Événements*. 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 inclue: Equals (Est égal), IsNull (Est nul), GreaterThan (Supérieur à), LessThan (Inférieur à), GreaterThanorEqual (Supérieur ou égal à), LessThanorEqual (Inférieur ou égal à), Like (Tel que), Not (Ne pas), Between (Entre), InList (Dans liste), And (Et), Or (Ou), Cast, BitwiseAnd (Au niveau du bit et), BitwiseOr (Au niveau du bit ou)	
Serveur	Contrôle Complex Event Filter (Filtre Evénements complexes)	Prend en charge les filtres d'*Événements* complexes, où "complexe" est défini comme prenant en charge les opérateurs de filtrage complexes (TypeOf) (TypeDe)	
Client	Contrôle Client Value Change (Modification de valeurs Client)	Utiliser le Jeu de *Services Éléments Surveillés* pour enregistrer les éléments propres aux modifications de la valeur d'*Attribut*. Utiliser CreateMonitoredItems (Créer Éléments Surveillés) pour enregistrer la séquence *Node/Attribute*. Définir l'intervalle d'échantillonnage, le Filtre de Bande Morte et le mode de mise en file d'attente appropriés. Utiliser le mode désactivation / activation de préférence au mode suppression et recréation d'un *Élément Surveillé*. Exécuter des opérations d'ensemble de préférence plutôt que des requêtes de service individuel pour réduire les volumes de communication.	
Client	Contrôle Client Deadband Filter (Filtre de Bande Morte Client)	Utilise les filtres de Bande Morte Absolus pour les abonnements.	
Client	Contrôle Client by Index (Client par Indice)	Utiliser IndexRange pour sélectionner un seul élément ou un ensemble d'éléments lorsque la valeur *Attribut* constitue une matrice.	
Client	Contrôle Client Aggregate Filter (Filtre Aggrégat Client)	Utilise des filtres Aggrégat pour les Abonnements.	
Client	Contrôle Client Events (Événements Client)	Utiliser le Jeu de *Services Eléments Surveillés* pour créer les *Éléments Surveillés* destinés aux notifications d'*Événements*.	
Client	Contrôle Client Event Filter (Filtre Événements Client)	Utiliser le filtre *Evénement* pour appeler le *Service* CreateMonitoredItems afin de filtrer les Événements souhaités et de sélectionner les colonnes à prévoir pour chaque *Notification* d'*Événement*.	

Catégorie	Titre	Description	Dérivation
Client	Contrôle Client Complex Event Filter (Filtre d'événements complexes Client)	Utilise les filtres d'*Événements* complexes	
Client	Contrôle Client Modify (Modification Client)	Utiliser le *Service* ModifyMonitoredItems pour modifier les paramètres de configuration. Utiliser le *Service* SetMonitoringMode pour désactiver / activer l'échantillonnage et/ou l'édition.	
Client	Contrôle Client Trigger (Déclenchement Client)	Utiliser le Modèle de déclenchement si certains éléments sont à consigner uniquement en cas de déclenchement de certains autres éléments. Utiliser le Mode de Surveillance approprié pour ces éléments. Utiliser le *Service* SetTriggering pour relier ces éléments à l'élément déclencheur.	

Le Jeu de *Services Abonnement* comprend plusieurs *Unités de Conformité* (voir Tableau 10). Les *Serveurs* peuvent fournir certains aspects de cette fonctionnalité, se reporter aux *Profils* catégorisés comme *Profils de Serveur* pour des informations détaillées. Les *Clients* peuvent prendre en charge certains aspects de cette fonctionnalité, se reporter aux *Profils* catégorisés comme *Profils de Client* pour des informations détaillées.

Tableau 10 – Services Abonnement

Catégorie	Titre	Description	Dérivation
Serveur	Abonnement de base	Prend en charge les *Services Abonnement* suivants: CreateSubscription, ModifySubscription, DeleteSubscriptions, Publish, Republish et SetPublishingMode (Créer Abonnement, Modifier Abonnement, Supprimer Abonnement, Éditer, Rééditer, Définir Mode d'édition).	
Serveur	Abonnement Minimum 1	Prend en charge au moins 1 Abonnement par *Session*. Il faut que ce nombre soit pris en charge pour toutes les sessions requises minimales.	
Serveur	Abonnement Minimum 02	Prend en charge au moins 2 Abonnements par *Session*. Il faut que ce nombre soit pris en charge pour au moins la moitié des sessions requises minimales.	
Serveur	Abonnement Minimum 05	Prend en charge au moins 5 Abonnements par *Session*. Il faut que ce nombre soit pris en charge pour au moins la moitié des sessions requises minimales.	
Serveur	Abonnement Édition Min 02	Prend en charge au moins 2 requêtes de *Service* Édition par *Session*. Il faut que ce nombre soit pris en charge pour toutes les sessions requises minimales. La prise en charge de réédition est facultative et il n'est pas nécessaire de fournir une file de retransmission de notification. Mais le service réédition doit être fourni et il retourne les résultats du niveau d'opération approprié.	
Serveur	Abonnement Édition Min 05	Prend en charge au moins 5 requêtes de *Service Édition* par *Session*. Il faut que ce nombre soit pris en charge pour au moins la moitié des sessions requises minimales. 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 Édition Min 10	Prend en charge au moins 10 requêtes de *Service Édition* par *session*. Il faut que ce nombre soit pris en charge pour au moins la moitié des sessions requises minimales. 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.	

Catégorie	Titre	Description	Dérivation
Serveur	Abonnement Politique d'élimination d'édition	Respecte la politique spécifiée d'élimination des requêtes de *Service* Édition. Si le nombre maximal des requêtes de *Service* Édition est dans une file d'attente et si une nouvelle requête de *Service* Édition se présente, il faut éliminer la requête d'édition la "plus ancienne" en renvoyant l'erreur appropriée.	
Serveur	Abonnement Transfert	Prend en charge le *Service* TransferSubscriptions (Abonnements de transfert) pour un transfert d'*Abonnement* d'une *Session* à l'autre.	
Client	Abonnement Client de base	Utiliser le Jeu de *Services Abonnement* et *d'Éléments Surveillés* comme moyen efficace de détecter les changements de valeurs d'*Attribut* et/ou de recevoir les occurrences d'*Événements*. Définir les intervalles d'édition appropriés et maintenir actives les notifications et la durée de vie totale d'*Abonnement*. Fournir un nombre suffisant de requêtes d'édition 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 Réédition Client	Évalue le nombre de séquences des *Notifications* afin de détecter les *Notifications* perdues. Utilise le Service Réédition 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 (Abonnements de transfert)	Le *Client* prend en charge les *Abonnements* de transfert d'autres *Clients*. Cette *Unité de Conformité* est utilisée comme partie des *Clients* redondants.	
Client	Abonnement Client Multiple	Utilise les multiples Abonnements pour réduire la charge utile des *Notifications* individuelles.	
Client	Abonnement Édition configurable Client	Transmet de multiples requêtes de *Service* Édition afin de s'assurer que le *Serveur* est toujours capable de transmettre des *Notifications*. Le nombre de requêtes parallèles de *Service* Édition par *Session* doit être configurable.	

5.3 Caractéristiques relatives au transport et à la communication

Le Tableau 11 décrit les *Unités de Conformité* relatives à la sécurité. Toutes ces *Unités de Conformité* 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 est du ressort de l'administrateur de déterminer

quelle *Unité de Conformité* est présentée dans une application donnée déployée dans le *Serveur* ou le *Client*.

Tableau 11 – Sécurité

Catégorie	Titre	Description	Dérivation
Sécurité	Sécurité Validation de certificat	Un Certificat est validé tel que spécifié dans l'IEC 62541-4. Ceci inclut, entre autres choses, l'examen de la structure et de la signature. Permet la suppression de certaines erreurs de validation par directive administrative.	
Sécurité	Aucune (None) Sécurité	Suite d'algorithmes NE fournissant AUCUN paramètre de sécurité. -> SymmetricSignatureAlgorithm – Non utilisé -> SymmetricEncryptionAlgorithm – Non utilisé -> AsymmetricSignatureAlgorithm – Non utilisé -> SymmetricKeyWrapAlgorithm – Non utilisé -> AsymmetricEncryptionAlgorithm – Non utilisé -> KeyDerivationAlgorithm – Non utilisé -> DerivedSignatureKeyLength – 0 L'utilisation de cette suite d'algorithmes doit pouvoir être activée ou désactivée par un administrateur.	
Sécurité	Aucune (None) Sécurité CreateSession ActivateSession	Lorsque SecurityPolicy=None, le Service CreateSession et ActivateSession permet une signature NULLE/vide et ne requiert ni *Certificats* d'application, ni Nonce.	
Sécurité	Sécurité de base 128Rsa15	Suite d'algorithmes qui utilise RSA15 comme Key-Wrap-algorithm (algorithme d'enveloppement à clé) et le bit 128 pour les algorithmes de cryptage. -> SymmetricSignatureAlgorithm – HmacSha1 – (http://www.w3.org/2000/09/xmldsig#hmac-sha1). -> SymmetricEncryptionAlgorithm – Aes128 – (http://www.w3.org/2001/04/xmlenc#aes128-cbc). -> AsymmetricSignatureAlgorithm – RsaSha1 – (http://www.w3.org/2000/09/xmldsig#rsa-sha1). -> AsymmetricKeyWrapAlgorithm – KwRsa15 – (http://www.w3.org/2001/04/xmlenc#rsa-1_5). -> AsymmetricEncryptionAlgorithm – Rsa15 – (http://www.w3.org/2001/04/xmlenc#rsa-1_5). -> KeyDerivationAlgorithm – PSha1 – (http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/dk/p_sha1). -> DerivedSignatureKeyLength – 128. -> MinAsymmetricKeyLength – 1024 -> MaxAsymmetricKeyLength – 2048 -> CertificateSignatureAlgorithm – Sha1	

Catégorie	Titre	Description	Dérivation
Sécurité	Sécurité de base 256	Suite d'algorithmes pour cryptage Bit 256, les algorithmes comprennent: -> SymmetricSignatureAlgorithm – HmacSha1 – (http://www.w3.org/2000/09/xmldsig#hmac-sha1). -> SymmetricEncryptionAlgorithm – Aes256 – (http://www.w3.org/2001/04/xmlenc#aes256-cbc). -> AsymmetricSignatureAlgorithm – RsaSha1 – (http://www.w3.org/2000/09/xmldsig#rsa-sha1). -> AsymmetricKeyWrapAlgorithm – KwRsaOaep – (http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p). -> AsymmetricEncryptionAlgorithm – RsaOaep – (http://www.w3.org/2001/04/xmlenc#rsa-oaep). -> KeyDerivationAlgorithm – PSha1 – (http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/dk/p_sha1). -> DerivedSignatureKeyLength – 192 -> MinAsymmetricKeyLength – 1024 -> MaxAsymmetricKeyLength – 2048 -> CertificateSignatureAlgorithm – Sha1	
Sécurité	Sécurité de base 256 Sha256	Suite d'algorithmes pour cryptage Bit 256, les algorithmes comprennent: -> SymmetricSignatureAlgorithm – Hmac_Sha256 – (http://www.w3.org/2000/09/xmldsig#hmac-sha256). -> SymmetricEncryptionAlgorithm – Aes256_CBC – (http://www.w3.org/2001/04/xmlenc#aes256-cbc). -> AsymmetricSignatureAlgorithm – Rsa_Sha256 – (http://www.w3.org/2000/09/xmldsig#rsa-sha256). -> AsymmetricKeyWrapAlgorithm – KwRsaOaep – (http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p). -> AsymmetricEncryptionAlgorithm – Rsa_Oaep – (http://www.w3.org/2001/04/xmlenc#rsa-oaep). -> KeyDerivationAlgorithm – PSHA256 – (http://docs.oasis-open.org/ws-sx/ws-secureconversation/200512/dk/p_sha256). -> DerivedSignatureKeyLength – 256 -> MinAsymmetricKeyLength – 2048 -> MaxAsymmetricKeyLength – 4096 -> CertificateSignatureAlgorithm – Sha256 La prise en charge de ce profil de sécurité peut nécessiter la prise en charge d'un second certificat d'instance d'application, avec une plus grande taille de clé. Les applications doivent prendre en charge de multiples Certificats d'Instance d'Application si c'est exigé par les Politiques de Sécurité prises en charge et utiliser le certificat exigé pour un point d'extrémité de sécurité donné.	

Catégorie	Titre	Description	Dérivation
Sécurité	Sécurité TLS Générale	Cette *Unité de Conformité* 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 1.1	La connexion est établie à l'aide de TLS 1.1. Il est nécessaire de configurer l'application pour empêcher les connexions TLS 1.0, à moins que la connexion TLS 1.0 n'utilise TLS_RSA_WITH_RC4_128_SHA comme décrit dans l'*Unité de Conformité* "Security TLS_RSA_WITH_RC4_128_SHA"	
Sécurité	Sécurité TLS_RSA_WITH_RC4_128_SHA	La connexion est établie à l'aide de TLS_RSA_WITH_RC4_128_SHA. Il est nécessaire de configurer l'application pour empêcher l'utilisation de suites de protocole basées sur la cryptanalyse (TLS 1.0).	
Sécurité	Sécurité TLS_RSA_WITH_AES_256_CBC_SHA256	La connexion est établie à l'aide de TLS_RSA_WITH_AES_256_CBC_SHA256, qui a MinAsymmetricKeyLength – 2048, MaxAsymmetricKeyLength – 4096, AsymmetricSignatureAlgorithm – RSA_SHA256. (TLS 1.2)	
Sécurité	Sécurité Encryption Required (cryptage exigé)	Le cryptage utilisant les algorithmes fournis dans la suite d'algorithme de sécurité est exigé.	
Sécurité	Sécurité Signing Required (signature exigée)	La signature utilisant les algorithmes fournis dans la suite d'algorithme de sécurité est exigée.	
Sécurité	Sécurité Time Synch (synchronisation temporelle) – Configuration	L'application prend en charge la configuration d'un décalage d'horloge acceptable.	
Sécurité	Sécurité Time Synch – NTP / OS Based support (synchronisation temporelle – prise en charge par NTP ou par système d'exploitation)	L'application prend en charge la synchronisation temporelle, soit par la mise en oeuvre du Protocole d'Heure Réseau (NTP – Network Time Protocol), soit par les caractéristiques d'un système d'exploitation normalisé.	
Sécurité	Sécurité Time Synch – UA based support (synchronisation temporelle – prise en charge par UA)	Une application utilise les réponses de l'horodatage d'en-tête fournies par une source connue configurée, telle qu'un *Serveur de Découverte*, afin de synchroniser le temps sur l'application et afin que cette synchronisation temporelle se produise régulièrement. L'utilisation de cette TimeSyncing (synchronisation temporelle) peut être configurée.	

Catégorie	Titre	Description	Dérivation
Sécurité	Administration sécurité	Permet la configuration des éléments suivants relatifs à la sécurité. * sélectionne la(les) politique(s) autorisée(s) d'identification de l'utilisateur (Nom/mot de passe de l'utilisateur ou X509 ou Kerberos ou Anonymous) * active/désactive la politique de sécurité "None" (Aucun) ou autres politiques de sécurité * active/désactive les points d'extrémité avec MessageSecurityMode (Mode Sécurité Message) SIGN (Signature) ou SIGNANDENCRYPT (Signature et Cryptage). * définit les autorités de certification admises * définit la méthode de réaction aux *Certificats* inconnus	
Sécurité	Administration sécurité – 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 défini dans l'IEC 62541-6.	
Sécurité	Administration Certificat de sécurité	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 ApplicationInstanceCertificate par défaut	Une application, une fois installée, a par défaut un ApplicationInstanceCertificate valide. L'ApplicationInstanceCertificate par défaut doit soit être créé en tant que partie de l'installation, soit les instructions d'installation décrivent de façon explicite le processus pour créer et appliquer un ApplicationInstanceCertificate par défaut à l'application.	
Sécurité	Sécurité – pas d'Authentification de l'Application	Le *Serveur* prend en charge la capacité à n'être configuré pour aucune authentification d'application, seulement l'authentification de l'utilisateur et le cryptage/la signature normal/e: – Configurer le serveur 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	Meilleure pratique – événements 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	Meilleure pratique – Gestion des alarmes	Il convient qu'un Serveur limite la fonctionnalité 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.	

Catégorie	Titre	Description	Dérivation
Sécurité	Meilleure pratique – nombres aléatoires	Tous les nombres aléatoires requis pour l'utilisation en toute sécurité de générateurs de nombres aléatoires basés sur une bibliothèque cryptographique appropriée.	
Sécurité	Meilleure pratique – 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 TR 62541-2 pour plus de détails).	
Sécurité	Meilleure pratique – 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 plates-formes qui prennent en charge plusieurs rôles administratifs (telles que Windows ou Linux).	
Sécurité	Meilleure pratique – 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 l'IEC 62541-6.	
Sécurité	Meilleure pratique – Client Evénements d'Audit	Le système de suivi d'Audit se connecte à un Serveur utilisant un Canal Sécurisé et avec les droits administratifs appropriés pour permettre un accès aux Événements d'Audit.	
Sécurité	Sécurité Mot de passe Nom d'utilisateur	Le Serveur prend en charge la(les) combinaison(s) Nom d'utilisateur/Mot de passe. Le cryptage du mot de passe avec l'algorithme fourni dans le UserNameIdentityToken est requis en l'absence de cryptage de message.	
Sécurité	Sécurité Utilisateur X509	Le Serveur prend en charge une paire de clés publique/privée pour l'identité de l'utilisateur. L'utilisation de cette caractéristique doit pouvoir 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. L'utilisation de cette caractéristique doit pouvoir être activée ou désactivée par un Administrateur. Un cryptage spécifique du IssuedToken est requis en l'absence de cryptage de message. L'utilisation de ce jeton est définie dans la Documentation de jeton Kerberos.	
Sécurité	Sécurité Utilisateur Windows IssuedToken Kerberos	Le *Serveur* prend en charge la mise en œuvre Windows des jetons Kerberos. Cette *Unité de Conformité* ne s'applique que si "Securité Utilisateur IssuedToken Kerberos" est pris en charge.	
Sécurité	Sécurité Utilisateur Anonymous	Le *Serveur* prend en charge l'accès Anonymous. L'utilisation de cette caractéristique doit pouvoir être activée ou désactivée par un Administrateur. L'accès Anonymous doit être désactivé par défaut.	
Sécurité	Sécurité Utilisateur Client IssuedToken Kerberos	Un *Client* utilise un jeton de *Serveur* Kerberos. Le cryptage spécifique du issuedToken est requis en l'absence de cryptage de message. L'utilisation de ce jeton est définie dans la documentation Kerberos.	

Catégorie	Titre	Description	Dérivation
Sécurité	Sécurité Utilisateur Client Windows IssuedToken Kerberos	Un *Client* utilise la mise en oeuvre Windows des jetons Kerberos. Cette *Unité de Conformité* ne s'applique que si le "Securité Utilisateur Client IssuedToken Kerberos" est pris en charge.	
Sécurité	Sécurité Nom d'Utilisateur Mot de Passe Client	Un *Client* utilise une combinaison Nom d'Utilisateur/Mot de Passe. Le cryptage du mot de passe avec l'algorithme fourni dans le UserNameIdentityToken est requis en l'absence de cryptage de message.	
Sécurité	Sécurité Utilisateur Client X509	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.	

Le Tableau 12 décrit les caractéristiques relatives au protocole et au codage qui peuvent être intégrées dans les profils. 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	Titre	Description	Dérivation
Serveur	Configuration de protocole	Permet l'administration des Points d'extrémité et du nombre de ports utilisé par ces derniers.	
Transport	Protocole TCP Binaire Sécurité UA	Prend en charge le protocole de transport TCP UA avec codage binaire UA et Conversation sécurisée UA.	
Transport	Protocole HTTPS avec Binaire UA	Prend en charge le protocole HTTPS avec codage binaire UA.	
Transport	Protocole HTTPS avec Soap	Prend en charge le protocole HTTPS avec codage Xml basé sur Soap.	
Transport	Protocole Soap Xml Sécurité WS	Prend en charge le protocole de transport "SOAP/HTTP" avec codage XML et Conversation sécurisée WS.	
Transport	Protocole Soap Binaire Sécurité WS	Prend en charge le protocole de transport "SOAP/HTTP" avec codage binaire UA et Conversation sécurisée WS.	

5.4 Modèle d'informations 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 des informations supplémentaires concernant 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	Titre	Description	Dérivation
Serveur	Info de Base – Structure principale	Le *Serveur* prend en charge l'*Objet Serveur,* les capacités du Serveur et la structure de l'*Espace d'adresses* OPC UA.	
Serveur	Info de Base – Capacités du Serveur	*Le Serveur* prend en charge l'édition de la limite du Serveur dans les Capacités du Serveur, y compris MaxArrayLength, MaxStringLength, MaxNodePerRead, MaxNodesPerWrite, MaxNodesPerSubscription et MaxNodesPerBrowse.	
Serveur	Info de Base – Événements Progrès	Le *Serveur* présente si la génération d'événements Progrès est prise en charge pour des appels de service longue durée, tels que HistoryRead (Lecture Historique) ou Query (Consultation). Si elle est indiquée comme prise en charge dans les Capacités du Serveur, alors les événements réels sont vérifiés.	
Serveur	Info de Base – diagnostics	Le *Serveur* prend en charge les *Objets* et *Variables* de diagnostic.	
Serveur	Info de Base – Statut système	Le *Serveur* prend en charge la génération de SystemStatusChangeEventType indiquant l'arrêt du *Serveur* (SourceNode=*Serveur*).	
Serveur	Info de Base – Système sous-jacent Statut système	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	Info 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	Info de Base – System Type	Le *Serveur* présente un Système Type avec des Types de données, Types de Références, *Types d'objets* et Types de Variables, y compris tous les types OPC UA (espace de nom 0) que le *Serveur* utilise, comme défini dans l'IEC 62541-5. Les éléments définis dans l'espace de nom 0 mais définis dans d'autres parties de la spécification sont soumis à l'essai comme partie des autres modèles d'informations.	
Serveur	Info de Base – Système type personnalisé	Le *Serveur* prend en charge la définition des *types d'Objet*, *Types de Variable*, *Types de Référence* ou Types de Données définis par l'utilisateur. La prise en charge de cette unité de conformité n'exige pas qu'un *Serveur* présente des types d'*Objet*, de *Variable*, de *Référence* ou de Données OPC UA à moins que le *Serveur* mette en œuvre des types Utilisateur. Si des types Utilisateur sont définis, la hiérarchie de types complète est également à présenter.	

Catégorie	Titre	Description	Dérivation
Serveur	Info de Base – Modification de modèle	Le *Serveur* prend en charge l'*Événement* ModelChange et la *Propriété* NodeVersion pour tous les *Nœuds* pour lesquels le serveur permet des modifications du Modèle.	
Serveur	Info 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 (paramètre fictif facultatif) ou MandatoryPlaceholder (paramètre fictif obligatoire).	
Serveur	Info 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, tel qu'un changement dans l'unité technique associé à une valeur.	
Serveur	Info de Base – EventQueueOverflowEventType	Le *Serveur* prend en charge l'EventQueueOverflowEventType comme défini dans l'IEC 62541-4.	
Serveur	Info de Base – OptionSet	Le *Serveur* prend en charge l'OptionSet *VariableType*.	
Serveur	Info de Base – ValueAsText	Le *Serveur* prend en charge la *Propriété* ValueAsText pour les Types de Données énumérés.	
Serveur	Info 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" (Codes pour les unités de mesure) de l'UN/CEFACT. Si une autre représentation EU est requise l'EUInformation.namespaceUri indique l'espace de nom alternatif.	
Serveur	Info 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	Info de Base – FileType Write	Le *Serveur* prend en charge l'*Objet* FileType, y compris l'écriture de fichiers, ainsi que le contrôle de l'accès de l'utilisateur à l'*Objet* FileType.	
Client	Info de Base – Client de base	Le *Client* utilise l'*Espace d'adresses* OPC UA défini. Accède ou permet l'accès aux informations du *Serveur* telles que l'état du *Serveur*, BuildInfo, les capacités, le Tableau des espaces de noms et le Modèle de Type.	
Client	Info de Base Client – Statut de Système	Le *Client* utilise SystemStatusChangeEventType pour détecter un arrêt du Serveur.	
Client	Info de Base Client – Événements progrès	Le *Client* utilise ProgressEvents, y compris la vérification de leur prise en charge.	
Client	Info de Base Client – Diagnostics	Le *Client* permet un accès interactif ou par programmation aux informations de diagnostic *du Serveur*.	

Catégorie	Titre	Description	Dérivation
Client	Info 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 Types de Données, *Types d'Objets* et Types de Variables personnalisés	
Client	Info de Base Client – Événement modification	Le *Client* traite les ModelChangeEvents afin de détecter les modifications de l'*Espace d'adresses* OPC UA du *Serveur* et prendre les mesures appropriées pour une modification donnée.	
Client	Info 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	Info 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	Info de Base Client – FileType Write	Le *Client* peut accéder à un *Objet* FileType pour transférer un fichier du *Client* au *Serveur*. Cela comprend des fichiers de grande taille.	

Le Tableau 14 décrit les éléments relatifs aux informations concernant le modèle de l'Espace d'adresses 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. Ceci inclut les *Facettes Serveur* qui décrivent les éléments présentés par un *Serveur* et les *Facettes Client* qui décrivent ce que consomme un *Client*.

Tableau 14 – Modèle de l'Espace d'adresses

Catégorie	Titre	Description	Dérivation
Serveur	Base de l'Espace d'adresses	Prend en charge les *Classes de Nœuds* avec leurs *Attributs* et leur comportement, tel que défini dans l'IEC 62541-3. Ceci inclut, par exemple, les attributs suivants: *Objet, Type d'Objet, Variable, Type de Variable, Références* et *Type de Données*	
Serveur	Événements de l'Espace d'adresses	Prend en charge les éléments de *l'Espace d'adresses* OPC UA afin de générer les notifications d'*Événements*. Ceci inclut au moins un *Nœud* avec un *Attribut EventNotifier* mis à Vrai (*Nœud de serveur*).	
Serveur	Types de données complexes de l'Espace d'adresses	Prend en charge les StructuredDataTypes avec un Dictionnaire de Données.	
Serveur	Méthode de l'Espace d'adresses	Prend en charge les *Nœuds de Méthode*.	
Serveur	Hiérarchie des Notifications de l'Espace d'adresses	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	Hiérarchie des sources de l'Espace d'adresses	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'adresses – WriteMask	Prend en charge 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'adresses – UserWriteMask	Prend en charge 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'adresses – UserWriteMask Multilevel	Prend en charge 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 noeuds dans le système.	
Serveur	Espace d'adresses – 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 *Variable* dans le système. Cela comprend une indication d'accès à l'Attribut Valeur en lecture, écriture, lecture Historique et écriture Historique.	

Catégorie	Titre	Description	Dérivation
Serveur	Espace d'adresses – Niveau d'accès utilisateur de base	Met en œuvre la sécurité du niveau d'accès de l'utilisateur pour les nœuds *Variable*, 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	Base de l'Espace d'adresses Client	Utilise et comprend les *Classes de Nœuds* avec leurs *Attributs* et leur comportement, tel que défini dans l'IEC 62541-3. Ceci inclut, par exemple, les attributs suivants: *Objet, Type d'Objet, Variable, Type de Variable, Références* et *Type de Données*. Ceci inclut de considérer les BrowseNames et String NodeIds comme sensibles à la casse.	
Client	Espace d'adresses – Type de Données complexes Client	Utilise et comprend les StructuredDataTypes arbitraires via le Dictionnaire de Données.	
Client	Espace d'adresses – Hiérarchie de Notification Client	Utilise la hiérarchie des *Nœuds d'Objet* qui sont des notifications afin de détecter des zones spécifiques où le *Client* peut s'abonner pour des Événements.	
Client	Espace d'adresses – Hiérarchie de Source Client	Détecte et utilise la hiérarchie des sources d'événement présentée pour des *Nœuds d'Objet* spécifiques qui sont des notifications d'événement.	

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

Tableau 15 – Accès aux données

Catégorie	Titre	Description	Dérivation
Serveur	Accès aux données DataItems	Fournit des *Variables* du DataItemType ou un de ses sous-types. Prend en charge les Codes de Statut spécifiés dans l'IEC 62541-8. La prise en charge des Propriétés facultatives (par exemple "InstrumentRange") doit être vérifiée au cours des essais de certification et figure 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 énumérée.	
Serveur	Accès aux données PercentDeadband	Prend en charge le filtre PercentDeadband lors de la surveillance des *Variables* AnalogItemType.	
Serveur	Accès aux données Changements sémantiques	Prend en charge les changements sémantiques des éléments AnalogItemType (*Propriété* EURange et/ou *Propriété* EngineeringUnits). Prend en charge les bits du Code de Statut des changements sémantiques le cas échéant.	
Serveur	Accès aux données TwoState (Deux États)	Prend en charge les *Variables* TwoStateDiscreteType avec les Propriétés correspondantes.	
Serveur	Accès aux données MultiState (États Multiples)	Prend en charge les *Variables* MultiStateDiscreteType avec les Propriétés correspondantes.	
Server	Accès aux données ArrayItemType	Fournit des *Variables* de l'ArrayItemType ou un de ses sous-types (YArrayItemType, XYArrayItemType, ImageArrayType, CubeArrayType et NDimensionArrayType). Les sous-types pris en charge sont énumérés. La prise en charge de ce type comprend la prise en charge de toutes les propriétés obligatoires, y compris AxisInformation.	
Server	Accès aux données Complex Number (numéro complexe)	Prend en charge le type de données Complex Number. Ce type de données est disponible pour tout type de variable n'ayant pas d'autre restriction explicite.	
Server	Accès aux données DoubleComplex Number (numéro complexe double)	Prend en charge le type de données DoubleComplex Number. Ce type de données est disponible pour tout type de variable n'ayant pas d'autre restriction explicite.	
Client	Accès aux données Client de base	Comprend les types de *Variables* Accès aux Données. Utilise les Propriétés normalisées, le cas échéant.	
Client	Accès aux données Deadband (Bande morte) Client	Utilise PercentDeadband pour filtrer les modifications de valeurs des *Variables* AnalogItemType.	

Catégorie	Titre	Description	Dérivation
Client	Accès aux données Changement sémantique Client	Reconnaît le bit de changement sémantique dans le Code de Statut tout en surveillant les éléments et en prenant les mesures appropriées. Généralement, il faut que le *Client* relise 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'informations *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'informations et les *Clients* qui traitent des *Alarmes* et *Conditions* utilisent ledit modèle.

Tableau 16 – Alarmes et Conditions

Catégorie	Titre	Description	Dérivation
Serveur	A & C Basic (de base)	Prend en charge le ConditionType du modèle *Alarme & Condition*	
Serveur	A & C Enable (active)	Prend en charge les Méthodes Enable et Disable.	
Serveur	A & C Refresh (rafraîchit)	Prend en charge la *Méthode* ConditionRefresh et le concept de rafraîchissement	
Serveur	A & C Instances	Prend en charge la présentation des *Conditions* A&C dans l'*Espace d'adresses*	
Serveur	A & C ConditionClasses (Classes de condition)	Prend en charge les multiples classes de *Condition* pour le groupement et le filtrage des *Alarmes*	
Serveur	A & C Acknowledge (Acquittement)	Prend en charge Acknowledge, inclut la *Méthode* Acknowledge et le type Acknowledgeable.	
Serveur	A & C Confirm (confirmation)	Prend en charge les *Conditions* de confirmation, inclut la méthode Confirm	
Serveur	A & C Comment (Commentaire)	Prend en charge Comments, inclut la *Méthode* AddComment.	
Serveur	A & C Alarm (Alarme)	Prend en charge la fonctionnalité *Alarme* de base, y compris les états actif/inactif	
Serveur	A & C Branch (branche)	Prend en charge les Branches *Alarme*, ce qui inclut les instances de *Condition* précédentes, c'est-à-dire les instances de condition autres que la condition courante qui exige toujours une certaine action de l'opérateur, telle qu'un acquittement ou un dialogue.	
Serveur	A & C Shelving (Ordonnancement)	Prend en charge le mode shelving, y compris les méthodes TimedShelve, OneShotShelve et Unshelve	
Serveur	A & C Exclusive Level (Niveau exclusif)	Prend en charge le type d'*Alarme* Niveau Exclusif	
Serveur	A & C Exclusive Limit (Limite exclusive)	Prend en charge les *Alarmes* Limite exclusive. Un *Serveur* qui prend en charge ce type d'alarme doit prendre en charge un des sous-types Level (Niveau), Deviation (Écart) ou RateofChange (Taux de variation).	

Catégorie	Titre	Description	Dérivation
Serveur	A & C Exclusive Deviation (Ecart exclusif)	Prend en charge le type d'*Alarme* Ecart exclusif	
Serveur	A & C Exclusive RateofChange (Taux de variation exclusif)	Prend en charge le type d'*Alarme* Taux de variation exclusif	
Serveur	A & C Non-Exclusive Limit (Limite non exclusive)	Prend en charge les *Alarmes* Limite Non Exclusive. Un *Serveur* qui prend en charge ce type d'alarme doit prendre en charge un des sous-types Level, Deviation ou RateofChange.	
Serveur	A & C Non-Exclusive Level (Niveau non exclusif)	Prend en charge le type d'*Alarme* Niveau Non Exclusif	
Serveur	A & C Non-Exclusive Deviation (Écart non exclusif)	Prend en charge le type d'*Alarme* Ecart Non Exclusif	
Serveur	A & C Non-Exclusive RateofChange (Taux de variation non exclusif)	Prend en charge le type d'*Alarme* Taux de Variation Non Exclusif	
Serveur	A & C Discrete (Tout ou rien)	Prend en charge les types d'*Alarme* Tout ou Rien	
Serveur	A & C Off Normal (non normalisé)	Prend en charge le type d'*Alarme* Non Normalisée	
Serveur	A & C Trip (Déclenchement)	Prend en charge le type d'*Alarme* Déclenchement	
Serveur	A & C Dialog (Dialogue)	Prend en charge DialogConditionType, y compris la *Méthode* Respond (Répondre)	
Serveur	A & E Wrapper Mapping (Conteneur de Correspondance)	Le *Serveur* utilise la correspondance COM A&E spécifiée dans l'IEC 62541-9:2012 pour mettre en correspondance les Evénements COM et les Evénements A&C. Ceci inclut la correspondance des Classes de *Condition*	
Client	A & C Basic Client (Client de base)	Utilise le Type de Condition de modèle *Alarme* & *Condition*	
Client	A & C Enable Client (Activer Client)	Utilise les Méthodes Enable et Disable.	
Client	A & C Refresh Client (Rafraîchissement Client)	Utilise la *Méthode* ConditionRefresh et le concept de rafraîchissement	
Client	A & C Instances Client (Instances Client)	Utilise les *Conditions* A&C présentées dans l'*AddressSpace*.	
Client	A & C ConditionClasses Client (Classes de condition Client)	Utilise les classes de *Condition* pour grouper les *Alarmes*.	
Client	A & C Acknowledge Client (acquittement client)	Utilise Acknowledge, y compris la *Méthode* Acknowledge et le type Acknowledgeable.	
Client	A & C Confirm Client (Confirmation Client)	Utilise les *Conditions* de confirmation, y compris la méthode Confirm	
Client	A & C Comment Client (Commentaire Client)	Utilise Comments, y compris la *Méthode* AddComment.	
Client	A & C Alarm Client (Alarme Client)	Utilise la fonctionnalité *Alarme* de base, y compris les états actif/inactif	

Catégorie	Titre	Description	Dérivation
Client	A & C Branch Client (Branche Client)	Utilise les branches d'*Alarme* comportant les Instances de *Conditions* précédentes, c'est-à-dire les instances de condition autres que la condition courante qui exige toujours une certaine action, telle qu'un acquittement ou une confirmation.	
Client	A & C Shelving Client (Ordonnancement Client)	Utilise le modèle shelving, y compris les méthodes TimedShelve, OneShotShelve et Unshelve.	
Client	A & C Exclusive Level Client (Niveau exclusif Client)	Utilise les *Alarmes* Niveau Exclusif comme défini	
Client	A & C Exclusive Limit Client (Limite exclusive Client)	Utilise les *Alarmes* Limite Exclusive. Exige l'utilisation d'au moins un des sous-types.	
Client	A & C Exclusive Deviation Client (Écart exclusif Client)	Utilise les *Alarmes* Ecart Exclusif	
Client	A & C Exclusive RateofChange (Taux de variation exclusif)	Utilise les *Alarmes* Taux de Variation Exclusif	
Client	A & C Non-Exclusive Level Client (Niveau non exclusif Client)	Utilise les *Alarmes* Niveau Non Exclusif	
Client	A & C Non-Exclusive Limit Client (Limite non exclusive Client)	Utilise les *Alarmes* Limite Non Exclusive. Exige l'utilisation d'au moins un des sous-types	
Client	A & C Non-Exclusive Deviation Client (Écart non exclusif Client)	Utilise les *Alarmes* Ecart Non Exclusif	
Client	A & C Non-Exclusive RateofChange Client (Taux de variation non exclusif Client)	Utilise les *Alarmes* Taux de Variation Non Exclusif	
Client	A & C Discrete Client (Tout ou Rien Client)	Utilise les types d'*Alarme* Tout ou Rien	
Client	A & C Off Normal Client (non normalisé client)	Utilise les types d'*Alarme* Non normalisé	
Client	A & C Trip Client (Déclenchement Client)	Utilise le type d'*Alarme* Déclenchement	
Client	A & C Dialog Client (Dialogue Client)	Utilise DialogConditionType, y compris la *Méthode* Respond	

Le Tableau 17 décrit les éléments relatifs au modèle d'informations 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'informations et les *Clients* qui utilisent les données historiques utilisent ledit modèle.

Tableau 17 – Accès à l'historique

Catégorie	Titre	Description	Dérivation
Serveur	Accès à l'historique Read Raw (lecture de données brutes)	Prise en charge générale de l'Accès à l'historique de base, de la lecture de données brutes utilisant la structure ReadRawModifiedDetails. Si l'intervalle de temps est spécifié avec un temps de départ, d'arrêt et un nombre de valeurs (au moins deux des trois paramètres doivent être fournis) et le fanion ReadModified est mis sur Faux.	
Serveur	Accès à l'historique Data Max Nodes Read Continuation Point (point de continuation lecture nœuds max de données)	Prend en charge suffisamment de points de continuation pour couvrir le nombre de points pris en charge indiqué dans le paramètre OperationLimits du *Serveur* MaxNodesPerHistoryReadData pour accéder aux données historiques.	
Serveur	Accès à l'historique Time Instance (Instance Temps)	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 Aggregates (Agrégats)	Prend en charge la lecture d'u ou plusieurs Agrégats de valeurs historiques de *Variables* à l'aide de la structure ReadProcessedDetails. Au moins un des Agrégats décrits dans l'IEC 62541-13 doit être pris en charge. La liste complète est indiquée dans le *Certificat de logiciel*.	
Serveur	Accès à l'historique Insert Value (Insérer Valeur)	Prend en charge l'insertion de valeurs historiques de *Variables*.	
Serveur	Accès à l'historique Delete Value (Supprimer Valeur)	Prend en charge la suppression de valeurs historiques de *Variables*.	
Serveur	Accès à l'historique Update Value (Mettre à jour Valeur)	Prend en charge la mise à jour de valeurs historiques de *Variables*.	
Serveur	Accès à l'historique Replace Value (Remplacer Valeur)	Prend en charge le remplacement de valeurs historiques de *Variables*.	
Serveur	Accès à l'historique Modified Values (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 vrai).	
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 lecture de données historiques brutes normalisée (ReadRawModifiedDetails). La saisie utilise la fonctionnalité de mise à jour (UpdateStructureDataDetails) historique normalisée.	

Catégorie	Titre	Description	Dérivation
Serveur	Accès à l'historique ServerTimestamp (Horodatage Serveur)	Prend en charge l'apport d'un ServerTimestamp (ainsi que le SourceTimestamp par défaut).	
Serveur	Accès à l'historique Structured Data Read Raw (Lecture de données brutes structurées)	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 considérée comme prenant en charge des données structurées génériques.	
Serveur	Accès à l'historique Structured Data Time Instance (Instance de temps 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 considérée comme prenant en charge des données structurées génériques.	
Serveur	Accès à l'historique Structured Data Insert (Insertion 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 considérée comme prenant en charge des données structurées génériques.	
Serveur	Accès à l'historique Structured Data Delete (Suppression 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 considérée comme prenant en charge des données structurées génériques.	
Serveur	Accès à l'historique Structured Data Update (Mise à jour 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 considérée comme prenant en charge des données structurées génériques.	
Serveur	Accès à l'historique Structured Data Replace (Remplacement Données Structurées)	Prend en charge le remplacement de données historiques structurées. La prise en charge de la structure pour une annotation n'est pas considérée comme prenant en charge des données structurées génériques.	
Serveur	Accès à l'historique Structured Data Read Modified (Lecture 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 vrai) pour les données structurées. La prise en charge de la structure pour une annotation n'est pas considérée comme prenant en charge des données structurées génériques.	
Serveur	Accès à l'historique Events (Événements)	Prend en charge la récupération d'Événements historiques à l'aide de la structure ReadEventDetails. Cela comprend la prise en charge du filtrage simple d'Événements. Les champs *Evénement* stockés sont spécifiques au serveur, mais au moins les champs obligatoires de BaseEventType sont requis.	

Catégorie	Titre	Description	Dérivation
Serveur	Accès à l'historique Event Max Events Read Continuation Point (Point de Continuation Lecture d'Événements Max)	Prend en charge suffisamment de points de continuation pour couvrir le nombre de lectures d'*Événement* prises en charge indiqué dans le paramètre OperationLimits du *Serveur* MaxNodesPerHistoryReadEvents pour l'Accès à l'historique *Événement*.	
Serveur	Accès à l'historique Insert Event (Insertion Evénement)	Prend en charge l'insertion d'Événements historiques.	
Serveur	Accès à l'historique Update Event (Mise à jour Evénement)	Prend en charge la mise à jour d'Événements historiques.	
Serveur	Accès à l'historique Replace Event (Remplacement Evénement)	Prend en charge le remplacement d'Événements historiques.	
Serveur	Accès à l'historique Delete Event (Suppression Evénement)	Prend en charge la suppression d'Événements historiques.	
Client	Accès à l'historique Client Browse (Navigation Client)	Utilise le Jeu de *Services* Vue pour découvrir des *Nœuds* avec des données historiques.	
Client	Accès à l'historique Client Read Raw (Lecture Données Brutes Client)	Utilise le *Service* HistoryRead pour lire des données historiques brutes à l'aide de la Structure ReadRawModifiedDetails (Fanion ReadModified mis à Faux).	
Client	Accès à l'historique Client Read Modified (Lecture Données Modifiées Client)	Utilise le *Service* HistoryRead pour lire des données historiques modifiées à l'aide de la Structure ReadRawModifiedDetails (Fanion ReadModified mis à Vrai).	
Client	Accès à l'historique Client Read Aggregates (Lecture Agrégats Client)	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. La liste complète des Agrégats utilisés par le *Client* est comprise dans les résultats de cette *Unité de Conformité*.	
Client	Accès à l'historique Client Structure Data Raw (Données Brutes Structurées Client)	Utilise le *Service* HistoryRead pour lire des données historiques brutes à l'aide de la Structure ReadRawModifiedDetails (Fanion ReadModified mis à Faux) pour les données structurées.	
Client	Accès à l'historique Client Structure Data Read Modified (Lecture Données Structurées Modifiées Client)	Utilise le *Service* HistoryRead pour lire des données historiques structurées modifiées à l'aide de la Structure ReadRawModifiedDetails (Fanion ReadModified mis à Vrai).	
Client	Accès à l'historique Client Structure Data Insert (Insertion Données Structurées Client)	Utilise le *Service* HistoryUpdate pour insérer des valeurs de données historiques pour les données structurées.	

Catégorie	Titre	Description	Dérivation
Client	Accès à l'historique Client Structure Data Delete (Suppression Données Structurées Client)	Utilise le *Service* HistoryUpdate pour supprimer des valeurs de données historiques pour les données structurées.	
Client	Accès à l'historique Client Structure Data Update (Mise à jour Données Structurées Client)	Utilise le *Service* HistoryUpdate pour mettre à jour des valeurs de données historiques pour les données structurées.	
Client	Accès à l'historique Client Structure Data Replace (Remplacement Données Structurées Client)	Utilise le *Service* HistoryUpdate pour remplacer des valeurs de données historiques pour les données structurées.	
Client	Accès à l'historique Client Structure Data Time Instance (Instance Temps Données Structurées Client)	Lit les données historiques à une instance spécifiée dans le temps pour les données structurées. Utilise la structure ReadAtTimeDetails.	
Client	Accès à l'historique Client Read Events (Lecture Événements Client)	Utilise le *Service* HistoryRead pour lire les données historiques *Evènement* à l'aide de la Structure ReadEventDetails.	
Client	Accès à l'historique Client Event Inserts (Insertions Événement Client)	Utilise le *Service* HistoryUpdate pour insérer des Événements historiques.	
Client	Accès à l'historique Client Event Updates (Mise à jour Événement Client)	Utilise le *Service* HistoryUpdate pour mettre à jour des Événements historiques.	
Client	Accès à l'historique Client Event Replaces (Remplacement Événement Client)	Utilise le *Service* HistoryUpdate pour remplacer des Événements historiques.	
Client	Accès à l'historique Client Event Deletes (Suppression Événement Client)	Utilise le *Service* HistoryUpdate pour supprimer des Événements historiques.	
Client	Accès à l'historique Client Data Insert (Insertion Données Client)	Utilise le *Service* HistoryUpdate pour insérer des valeurs de données historiques.	
Client	Accès à l'historique Client Data Delete (Suppression Données Client)	Utilise le *Service* HistoryUpdate pour supprimer des valeurs de données historiques	
Client	Accès à l'historique Client Data Update (Mise à jour Données Client)	Utilise le *Service* HistoryUpdate pour mettre à jour des valeurs de données historiques.	

Catégorie	Titre	Description	Dérivation
Client	Accès à l'historique Client Data Replace (Remplacement Données Client)	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 lecture de données historiques brutes normalisée (ReadRawModifiedDetails). La saisie utilise la fonctionnalité de mise à jour (UpdateStructureDataDetails) historique normalisée.	
Client	Accès à l'historique Client Time Instance (Instance Temps Client)	Lit les données historiques à une instance spécifiée dans le temps à l'aide de la structure ReadAtTimeDetails.	
Client	Accès à l'historique Client Server Timestamp (Horodatage Serveur Client)	Utilise le ServerTimestamp (ainsi que le SourceTimestamp par défaut), s'il est fourni par le *Serveur*.	

Le Tableau 18 décrit les éléments relatifs à l'Aggrégat qui peuvent être intégrés dans les profils. Les *Servers* prenant en charge les Aggrégats présentent cette fonctionnalité et les *Clients* qui utilisent les Aggrégats mettent en œuvre une partie de la fonctionnalité.

Tableau 18 – Agrégats

Catégorie	Titre	Description	Dérivation
Serveur	Agrégat master configuration (configuration maître)	Prend en charge au moins un *Objet* AggregateConfigurationType maître comme partie de la configuration du *Serveur*.	
Serveur	Agrégat optional configuration (configuration facultative)	Prend en charge au moins un *Objet* AggregateConfigurationType facultatif. Des *Objets* AggregateConfigurationType facultatifs se rencontrent à différents niveaux de l'*Objet* AggregateConfigurationType maître.	
Serveur	Agrégat – Interpolative (Interpolatif)	Prend en charge l'Aggrégat Interpolative pour l'Accès à l'historique.	
Serveur	Agrégat – Average (Moyen)	Prend en charge l'Aggrégat Average pour l'Accès à l'historique.	
Serveur	Agrégat – TimeAverage (Temps Moyen)	Prend en charge l'Aggrégat TimeAverage pour l'Accès à l'historique.	
Serveur	Agrégat – TimeAverage2	Prend en charge l'Aggrégat TimeAverage2 pour l'Accès à l'historique.	
Serveur	Agrégat – Total	Prend en charge l'Aggrégat Total pour l'Accès à l'historique.	
Serveur	Agrégat – Total2	Prend en charge l'Aggrégat Total2 pour l'Accès à l'historique.	
Serveur	Agrégat – Minimum	Prend en charge l'Aggrégat Minimum pour l'Accès à l'historique	
Serveur	Agrégat – MinimumActualTime (Temps réel minimal)	Prend en charge l'Aggrégat MinimumActualTime pour l'Accès à l'historique.	
Serveur	Agrégat – Minimum2	Prend en charge l'Aggrégat Minimum2 pour l'Accès à l'historique.	
Serveur	Agrégat – MinimumActualTime2 (Temps réel minimal)	Prend en charge l'Aggrégat MinimumActualTime2 pour l'Accès à l'historique.	
Serveur	Agrégat – Maximum	Prend en charge l'Aggrégat Maximum pour l'Accès à l'historique.	
Serveur	Agrégat – MaximumActualTime (Temps réel maximal)	Prend en charge l'Aggrégat MaximumActualTime pour l'Accès à l'historique.	
Serveur	Agrégat – Maximum2	Prend en charge l'Aggrégat Maximum2 pour l'Accès à l'historique.	
Serveur	Agrégat – MaximumActualTime2 (Temps réel maximal)	Prend en charge l'Aggrégat MaximumActualTime2 pour l'Accès à l'historique.	
Serveur	Agrégat – Range (Gamme)	Prend en charge l'Aggrégat Range pour l'Accès à l'historique.	
Serveur	Agrégat – Range2 (Gamme)	Prend en charge l'Aggrégat Range2 pour l'Accès à l'historique.	
Serveur	Agrégat – Count (Compte)	Prend en charge l'Aggrégat Count pour l'Accès à l'historique.	
Serveur	Agrégat – DurationInStateZero (Durée à l'état zéro)	Prend en charge l'Aggrégat DurationInStateZero pour l'Accès à l'historique.	

Catégorie	Titre	Description	Dérivation
Serveur	Agrégat – DurationInStateNonZero (Durée à l'état non zéro)	Prend en charge l'Aggrégat DurationInStateNonZero pour l'Accès à l'historique.	
Serveur	Agrégat – NumberOfTransitions (Nombre de Transitions)	Prend en charge l'Aggrégat NumberOfTransitions pour l'Accès à l'historique.	
Serveur	Aggregate – Start (Démarrage)	Prend en charge l'Aggrégat Start pour l'Accès à l'historique.	
Serveur	Agrégat – StartBound (Prêt à démarrer)	Prend en charge l'Aggrégat StartBound pour l'Accès à l'historique.	
Serveur	Agrégat – End (Fin)	Prend en charge l'Aggrégat End pour l'Accès à l'historique.	
Serveur	Agrégat – EndBound (Prêt à finir)	Prend en charge l'Aggrégat EndBound pour l'Accès à l'historique.	
Serveur	Agrégat – Delta	Prend en charge l'Aggrégat Delta pour l'Accès à l'historique.	
Serveur	Agrégat – DeltaBounds (Prêt pour Delta)	Prend en charge l'Aggrégat DeltaBounds pour l'Accès à l'historique.	
Serveur	Agrégat – DurationGood (Durée Bon)	Prend en charge l'Aggrégat DurationGood pour l'Accès à l'historique.	
Serveur	Agrégat – DurationBad (Durée Mauvais)	Prend en charge l'Aggrégat DurationBad pour l'Accès à l'historique.	
Serveur	Agrégat – PercentGood (Pourcentage Bon)	Prend en charge l'Aggrégat PercentGood pour l'Accès à l'historique.	
Serveur	Agrégat – PercentBad (Pourcentage Mauvais)	Prend en charge l'Aggrégat PercentBad pour l'Accès à l'historique.	
Serveur	Agrégat – WorstQuality (Qualité la plus défavorable)	Prend en charge l'Aggrégat WorstQuality pour l'Accès à l'historique.	
Serveur	Agrégat – WorstQuality2 (Qualité la plus défavorable)	Prend en charge l'Aggrégat WorstQuality2 pour l'Accès à l'historique.	
Serveur	Agrégat – AnnotationCount (Compte Annotation)	Prend en charge l'Aggrégat AnnotationCount pour l'Accès à l'historique.	
Serveur	Agrégat – StandardDeviationSample (Échantillon Ecart Type)	Prend en charge l'Aggrégat StandardDeviationSample pour l'Accès à l'historique.	
Serveur	Agrégat – VarianceSample (Echantillon Variance)	Prend en charge l'Aggrégat VarianceSample pour l'Accès à l'historique.	
Serveur	Agrégat – StandardDeviationPopulation (Population Ecart Type)	Prend en charge l'Aggrégat StandardDeviationPopulation pour l'Accès à l'historique.	
Serveur	Agrégat – VariancePopulation (Population Variance)	Prend en charge l'Aggrégat VariancePopulation pour l'Accès à l'historique.	

Catégorie	Titre	Description	Dérivation
Serveur	Agrégat – Custom (personnalisé)	Le *Serveur* prend en charge les Aggrégats personnalisés pour l'Accès à l'historique pour lesquels aucun essai normalisé n'est défini. Ces Aggrégats sont indiqués comme non soumis à l'essai par cette *Unité de Conformité*.	
Serveur	Agrégat Abonnement – Filter (Filtre)	Prend en charge l'Agrégat Filtres Abonnement qui nécessite qu'au moins un des Aggrégats définis soit pris en charge, comme défini dans l'IEC 62541-13.	
Serveur	Agrégat Abonnement – Interpolative (Interpolatif)	Prend en charge le filtre abonnement pour l'Aggrégat Interpolative.	
Serveur	Agrégat Abonnement – Average (Moyen)	Prend en charge le filtre abonnement pour l'Aggrégat Average.	
Serveur	Agrégat Abonnement – TimeAverage (Temps Moyen)	Prend en charge le filtre abonnement pour l'Aggrégat TimeAverage.	
Serveur	Agrégat Abonnement – TimeAverage2 (Temps Moyen)	Prend en charge le filtre abonnement pour l'Aggrégat TimeAverage2.	
Serveur	Agrégat Abonnement – Total	Prend en charge le filtre abonnement pour l'Aggrégat Total.	
Serveur	Agrégat Abonnement – Total2	Prend en charge le filtre abonnement pour l'Aggrégat Total2.	
Serveur	Agrégat Abonnement – Minimum	Prend en charge le filtre abonnement pour l'Aggrégat Minimum.	
Serveur	Agrégat Abonnement – MinimumActualTime (Temps réel minimal)	Prend en charge le filtre abonnement pour l'Aggrégat MinimumActualTime.	
Serveur	Agrégat Abonnement – Minimum2	Prend en charge le filtre abonnement pour l'Aggrégat Minimum2.	
Serveur	Agrégat Abonnement – MinimumActualTime2 (Temps réel minimal)	Prend en charge le filtre abonnement pour l'Aggrégat MinimumActualTime2.	
Serveur	Agrégat Abonnement – Maximum	Prend en charge le filtre abonnement pour l'Aggrégat Maximum.	
Serveur	Agrégat Abonnement – MaximumActualTime (Temps réel maximal)	Prend en charge le filtre abonnement pour l'Aggrégat MaximumActualTime.	
Serveur	Agrégat Abonnement – Maximum2	Prend en charge le filtre abonnement pour l'Aggrégat Maximum2.	
Serveur	Agrégat Abonnement – MaximumActualTime2 (Temps réel maximal)	Prend en charge le filtre abonnement pour l'Aggrégat MaximumActualTime2.	
Serveur	Agrégat Abonnement – Range (Gamme)	Prend en charge le filtre abonnement pour l'Aggrégat Range.	
Serveur	Agrégat Abonnement – Range2 (Gamme)	Prend en charge le filtre abonnement pour l'Aggrégat Range2.	
Serveur	Agrégat Abonnement – Count (Compte)	Prend en charge le filtre abonnement pour l'Aggrégat Count.	

Catégorie	Titre	Description	Dérivation
Serveur	Agrégat Abonnement – DurationInStateZero (Durée à l'État Zéro)	Prend en charge le filtre abonnement pour l'Aggrégat DurationInStateZero.	
Serveur	Agrégat Abonnement – DurationInStateNonZero (Durée à l'État non Zéro)	Prend en charge le filtre abonnement pour l'Aggrégat DurationInStateNonZero.	
Serveur	Agrégat Abonnement – NumberOfTransitions (Nombre de Transitions)	Prend en charge le filtre abonnement pour l'Aggrégat NumberOfTransitions.	
Serveur	Agrégat Abonnement – Start (Démarrage)	Prend en charge le filtre abonnement pour l'Aggrégat Start.	
Serveur	Agrégat Abonnement – StartBound (Prêt à démarrer)	Prend en charge le filtre abonnement pour l'Aggrégat StartBound.	
Serveur	Agrégat Abonnement – End (Fin)	Prend en charge le filtre abonnement pour l'Aggrégat End.	
Serveur	Agrégat Abonnement – EndBound (Prêt à finir)	Prend en charge le filtre abonnement pour l'Aggrégat EndBound.	
Serveur	Agrégat Abonnement – Delta	Prend en charge le filtre abonnement pour l'Aggrégat Delta.	
Serveur	Agrégat Abonnement – DeltaBounds (Prêt pour Delta)	Prend en charge le filtre abonnement pour l'Aggrégat DeltaBounds.	
Serveur	Agrégat Abonnement – DurationGood (Durée Bon)	Prend en charge le filtre abonnement pour l'Aggrégat DurationGood.	
Serveur	Agrégat Abonnement – DurationBad (Durée Mauvais)	Prend en charge le filtre abonnement pour l'Aggrégat DurationBad.	
Serveur	Agrégat Abonnement – PercentGood (Pourcentage Bon)	Prend en charge le filtre abonnement pour l'Aggrégat PercentGood.	
Serveur	Agrégat Abonnement – PercentBad (Pourcentage Mauvais)	Prend en charge le filtre abonnement pour l'Aggrégat PercentBad.	
Serveur	Agrégat Abonnement – WorstQuality (Qualité la plus défavorable)	Prend en charge le filtre abonnement pour l'Aggrégat WorstQuality.	
Serveur	Agrégat Abonnement – WorstQuality2 (Qualité la plus défavorable)	Prend en charge le filtre abonnement pour l'Aggrégat WorstQuality2.	
Serveur	Agrégat Abonnement – AnnotationCount (Compte Annotation)	Prend en charge le filtre abonnement pour l'Aggrégat AnnotationCount.	
Serveur	Agrégat Abonnement – StandardDeviationSample (Echantillon Ecart Type)	Prend en charge le filtre abonnement pour l'Aggrégat StandardDeviationSample.	
Serveur	Agrégat Abonnement – VarianceSample (Echantillon Variance)	Prend en charge le filtre abonnement pour l'Aggrégat VarianceSample.	
Serveur	Agrégat Abonnement – StandardDeviationPopulation (Echantillon Ecart Type)	Prend en charge le filtre abonnement pour l'Aggrégat StandardDeviationPopulation.	
Serveur	Agrégat Abonnement – VariancePopulation (Population Variance)	Prend en charge le filtre abonnement pour l'Aggrégat VariancePopulation.	

Catégorie	Titre	Description	Dérivation
Serveur	Agrégat Abonnement – Custom (Personnalisé)	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 *Unité de Conformité*.	
Client	Agrégat – Client Usage (Utilisation Client)	Utilise l'Accès à l'historique à l'Agrégat qui nécessite qu'au moins un des Agrégats définis soit pris en charge comme défini dans l'IEC 62541-13.	
Client	Agrégat – Client Interpolative (Interpolatif Client)	Utilise l'Accès à l'historique à l'Agrégat Interpolatif.	
Client	Agrégat – Client Average (Moyen Client)	Utilise l'Accès à l'historique à l'Agrégat Average.	
Client	Agrégat – Client TimeAverage (Temps Moyen Client)	Utilise l'Accès à l'historique à l'Agrégat TimeAverage.	
Client	Agrégat – Client TimeAverage2 (Temps Moyen Client)	Utilise l'Accès à l'historique à l'Agrégat TimeAverage2.	
Client	Agrégat – Client Total	Utilise l'Accès à l'historique à l'Agrégat Total.	
Client	Agrégat – Client Total2	Utilise l'Accès à l'historique à l'Agrégat Total2.	
Client	Agrégat – Client Minimum (Minimum Client)	Utilise l'Accès à l'historique à l'Agrégat Minimum.	
Client	Agrégat – Client MinimumActualTime (Temps réel Minimal Client)	Utilise l'Accès à l'historique à l'Agrégat MinimumActualTime.	
Client	Agrégat – Client Minimum2 (Minimum Client)	Utilise l'Accès à l'historique à l'Agrégat Minimum2.	
Client	Agrégat – Client MinimumActualTime2 (Temps réel Minimal Client)	Utilise l'Accès à l'historique à l'Agrégat MinimumActualTime2.	
Client	Agrégat – Client Maximum (Maximum Client)	Utilise l'Accès à l'historique à l'Agrégat Maximum.	
Client	Agrégat – Client MaximumActualTime	Utilise l'Accès à l'historique à l'Agrégat MaximumActualTime.	
Client	Agrégat – Client Maximum2 (Maximum Client)	Utilise l'Accès à l'historique à l'Agrégat Maximum2.	
Client	Agrégat – Client MaximumActualTime2 (Temps réel Maximal Client)	Utilise l'Accès à l'historique à l'Agrégat MaximumActualTime2.	
Client	Agrégat – Client Range (Gamme Client)	Utilise l'Accès à l'historique à l'Agrégat Range.	
Client	Agrégat – Client Range2 (Gamme Client)	Utilise l'Accès à l'historique à l'Agrégat Range2.	
Client	Agrégat – Client Count (Compte Client)	Utilise l'Accès à l'historique à l'Agrégat Count.	
Client	Agrégat – Client DurationInStateZero (Durée à l'État Zéro)	Utilise l'Accès à l'historique à l'Agrégat DurationInStateZero.	
Client	Agrégat – Client DurationInStateNonZero (Durée à l'État non Zéro)	Utilise l'Accès à l'historique à l'Agrégat DurationInStateNonZero.	
Client	Agrégat – Client NumberOfTransitions (Nombre de Transitions	Utilise l'Accès à l'historique à l'Agrégat NumberOfTransitions.	

Catégorie	Titre	Description	Dérivation
Client	Aggregate – Client Start (Démarrage Client)	Utilise l'Accès à l'historique à l'Agrégat Start.	
Client	Agrégat – Client StartBound (Prêt à démarrer Client)	Utilise l'Accès à l'historique à l'Agrégat StartBound.	
Client	Agrégat – Client End (Fin Client)	Utilise l'Accès à l'historique à l'Agrégat End.	
Client	Agrégat – Client EndBound (Prêt à finir Client)	Utilise l'Accès à l'historique à l'Agrégat EndBound.	
Client	Agrégat – Client Delta (Delta Client)	Utilise l'Accès à l'historique à l'Agrégat Delta.	
Client	Agrégat – Client DeltaBounds (Prêt pour Delta Client)	Utilise l'Accès à l'historique à l'Agrégat DeltaBounds.	
Client	Agrégat – Client DurationGood (Durée Bon Client)	Utilise l'Accès à l'historique à l'Agrégat DurationGood.	
Client	Agrégat – Client DurationBad (Durée Mauvais Client)	Utilise l'Accès à l'historique à l'Agrégat DurationBad.	
Client	Agrégat – Client PercentGood (Pourcentage Bon Client)	Utilise l'Accès à l'historique à l'Agrégat PercentGood.	
Client	Agrégat – Client PercentBad (Pourcentage Mauvais Client)	Utilise l'Accès à l'historique à l'Agrégat PercentBad.	
Client	Agrégat – Client WorstQuality (Qualité la plus défavorable Client)	Utilise l'Accès à l'historique à l'Agrégat WorstQuality.	
Client	Agrégat – Client WorstQuality2 (Qualité la plus défavorable Client)	Utilise l'Accès à l'historique à l'Agrégat WorstQuality2.	
Client	Agrégat – Client AnnotationCount (Compte Annotation Client)	Utilise l'Accès à l'historique à l'Agrégat AnnotationCount.	
Client	Agrégat – Client StandardDeviationSample (Échantillon Écart Type Client)	Utilise l'Accès à l'historique à l'Agrégat StandardDeviationSample.	
Client	Agrégat – Client VarianceSample (Échantillon Variance Client)	Utilise l'Accès à l'historique à l'Agrégat VarianceSample.	
Client	Agrégat – Client StandardDeviationPopulation (Population Écart Type Client)	Utilise l'Accès à l'historique à l'Agrégat StandardDeviationPopulation.	
Client	Agrégat – Client VariancePopulation (Population Variance Client)	Utilise l'Accès à l'historique à l'Agrégat VariancePopulation.	
Client	Agrégat – Client Custom Aggregates (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	Agrégat Abonnement – Client Filter (Filtre Client)	S'abonne à des données utilisant des filtres Agrégat qui nécessitent qu'au moins un des Agrégats définis dans l'IEC 62541-13 soit pris en charge.	

Catégorie	Titre	Description	Dérivation
Client	Agrégat Abonnement – Client Interpolative (Interpolatif Client)	S'abonne à des données utilisant le filtre Agrégat Interpolative.	
Client	Agrégat Abonnement – Client Average (Moyen Client)	S'abonne à des données utilisant le filtre Agrégat Average.	
Client	Agrégat Abonnement – Client TimeAverage (Temps Moyen Client)	S'abonne à des données utilisant le filtre Agrégat TimeAverage.	
Client	Agrégat Abonnement – Client TimeAverage2 (Temps Moyen Client)	S'abonne à des données utilisant le filtre Agrégat TimeAverage2.	
Client	Agrégat Abonnement – Client Total	S'abonne à des données utilisant le filtre Agrégat Total.	
Client	Agrégat Abonnement – Client Total2	S'abonne à des données utilisant le filtre Agrégat Total2.	
Client	Agrégat Abonnement – Client Minimum	S'abonne à des données utilisant le filtre Agrégat Minimum.	
Client	Agrégat Abonnement – Client MinimumActualTime (Temps Réel Minimal Client)	S'abonne à des données utilisant le filtre Agrégat MinimumActualTime.	
Client	Agrégat Abonnement – Client Minimum2	S'abonne à des données utilisant le filtre Agrégat Minimum2.	
Client	Agrégat Abonnement – Client MinimumActualTime2 (Temps Réel Minimal Client)	S'abonne à des données utilisant le filtre Agrégat MinimumActualTime2.	
Client	Agrégat Abonnement – Client Maximum	S'abonne à des données utilisant le filtre Agrégat Maximum.	
Client	Agrégat Abonnement – Client MaximumActualTime (Temps Réel Maximal Client)	S'abonne à des données utilisant le filtre Agrégat MaximumActualTime.	
Client	Agrégat Abonnement – Client MaximumActualTime2 (Temps Réel Maximal Client)	S'abonne à des données utilisant le filtre Agrégat MaximumActualTime2.	
Client	Agrégat Abonnement – Client Maximum2	S'abonne à des données utilisant le filtre Agrégat Maximum2.	
Client	Agrégat Abonnement – Client Range (Gamme Client)	S'abonne à des données utilisant le filtre Agrégat Range.	
Client	Agrégat Abonnement – Client Range2 (Gamme Client)	S'abonne à des données utilisant le filtre Agrégat Range2.	
Client	Agrégat Abonnement – Client Count (Compte Client)	S'abonne à des données utilisant le filtre Agrégat Count.	
Client	Agrégat Abonnement – Client DurationInStateZero (Durée à l'État Zéro Client)	S'abonne à des données utilisant le filtre Agrégat DurationInStateZero.	
Client	Agrégat Abonnement – Client DurationInStateNonZero (Durée à l'État non Zéro Client)	S'abonne à des données utilisant le filtre Agrégat DurationInStateNonZero.	
Client	Agrégat Abonnement – Client NumberOfTransition (Nombre de Transitions Client)	S'abonne à des données utilisant le filtre Agrégat NumberOfTransitions.	
Client	Agrégat Abonnement – Client Start (Démarrage Client)	S'abonne à des données utilisant le filtre Agrégat Start.	

Catégorie	Titre	Description	Dérivation
Client	Agrégat Abonnement – Client StartBound (Prêt à démarrer Client)	S'abonne à des données utilisant le filtre Agrégat StartBound.	
Client	Agrégat Abonnement – Client End (Fin Client)	S'abonne à des données utilisant le filtre Agrégat End.	
Client	Agrégat Abonnement – Client EndBound (Prêt à finir Client)	S'abonne à des données utilisant le filtre Agrégat EndBound.	
Client	Agrégat Abonnement – Client Delta	S'abonne à des données utilisant le filtre Agrégat Delta.	
Client	Agrégat Abonnement – Client DeltaBounds (Prêt pour Delta Client)	S'abonne à des données utilisant le filtre Agrégat DeltaBounds.	
Client	Agrégat Abonnement – Client DurationGood (Durée Bon Client)	S'abonne à des données utilisant le filtre Agrégat DurationGood.	
Client	Agrégat Abonnement – Client DurationBad (Durée Mauvais Client)	S'abonne à des données utilisant le filtre Agrégat DurationBad.	
Client	Agrégat Abonnement – Client PercentGood (Pourcentage Bon Client)	S'abonne à des données utilisant le filtre Agrégat PercentGood.	
Client	Agrégat Abonnement – Client PercentBad (Pourcentage Mauvais Client)	S'abonne à des données utilisant le filtre Agrégat PercentBad.	
Client	Agrégat Abonnement – Client WorstQuality (Qualité la plus défavorable Client)	S'abonne à des données utilisant le filtre Agrégat WorstQuality.	
Client	Agrégat Abonnement – Client WorstQuality2 (Qualité la plus défavorable Client)	S'abonne à des données utilisant le filtre Agrégat WorstQuality2.	
Client	Agrégat Abonnement – Client AnnotationCount (Compte Annotation Client)	S'abonne à des données utilisant le filtre Agrégat AnnotationCount.	
Client	Agrégat Abonnement – Client StandardDevSample (Échantillon Écart Type Client)	S'abonne à des données utilisant le filtre Agrégat StandardDeviationSample.	
Client	Agrégat Abonnement – Client VarianceSample (Échantillon Variance Client)	S'abonne à des données utilisant le filtre Agrégat VarianceSample.	
Client	Agrégat Abonnement – Client StandardDevPopulation (Population Écart Type Client)	S'abonne à des données utilisant le filtre Agrégat StandardDeviationPopulation.	
Client	Agrégat Abonnement – Client VariancePopulation (Population Variance Client)	S'abonne à des données utilisant le filtre Agrégat VariancePopulation.	
Client	Agrégat Abonnement – Client Custom Aggregates (Agrégats Personnalisés 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 des 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* avec des capacités de ressources moindres ne peuvent 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	Titre	Description	Dérivation
Serveur	Base d'audit	Prend en charge les AuditEvents (Événements d'audit). La liste des Événements d'audit pris en charge doit être vérifiée au cours des essais de certification et figure sur le *Certificat de logiciel*. Les Événements d'audit de base sont définis dans l'IEC 62541-3 et dans l'IEC 62541-5.	
Client	Auditing Client Audit ID (Audit ID de l'audit Client)	Le *Client* prend en charge la génération des ids AuditEvents (identificateurs des Événements d'audit) et les fournit aux *Serveurs*.	
Client	Auditing Client Subscribes (Audit de l'Abonnement Client)	Le *Client* prend en charge l'abonnement aux Événements d'audit, 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 *Unités de Conformité* appropriées basées sur le type de redondance correspondant. Les *Clients* capables de gérer la redondance prennent en charge les *Unités de Conformité* appropriées basées sur le type de redondance pris en charge.

Tableau 20 – Redondance

Catégorie	Titre	Description	Dérivation
Serveur	Redondance Serveur	Prend en charge la redondance basée 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 reprendre la redondance d'un autre *Client* (exige une certaine communication hors bande)	
Client	Redondance Commutateur Client	Les *Clients* qui prennent en charge cette *Unité de Conformité* surveillent le statut de redondance dans le cas des *Serveurs* à redondance non transparente et commutent sur le *Serveur* auxiliaire lorsqu'ils identifient un changement dans le statut du serveur.	

5.5 Divers

Le Tableau 21 suivant décrit diverses *Unités de Conformité.*

Chaque tableau comporte une liste de la *Catégorie de profil* à laquelle appartient une *Unité de Conformité*, le titre et la description de l'*Unité de Conformité,* ainsi qu'une colonne qui indique si l'*Unité de Conformité* est issue d'une autre *Unité de Conformité*. Ce type d'*Unité de Conformité* inclut tous les essais correspondant au parent auxquels s'ajoute(nt) un ou

plusieurs Cas d'Essais supplémentaires. Ces Cas d'Essai peuvent uniquement limiter davantage les Cas d'Essai existants.

Tableau 21 – Divers

Catégorie	Titre	Description	Dérivation
Client, Serveur	Documentation – Supported Profiles (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.	
Client, Serveur	Documentation – Multiple Languages (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, Serveur	Documentation – Users Guide (Guide Utilisateur)	L'application comprend la documentation qui décrit la fonctionnalité disponible fournie par l'application. Pour les Serveurs, elle comprend un résumé de toutes les fonctionnalités assurées par le Serveur.	
Client, Serveur	Documentation – On-line (En ligne)	La documentation fournie par l'application est disponible au format électronique comme partie de l'application. La documentation électronique peut être une page WEB, un document installé ou un CD/DVD, mais quelle qu'elle soit, on peut y accéder depuis l'application ou depuis un lien installé avec l'application.	
Client, 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.	
Client, Serveur	Documentation – Trouble Shooting Guide (Guide de recherche de pannes)	L'application comprend la documentation qui décrit les problèmes typiques qu'un utilisateur peut rencontrer et 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 pouvant 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.	

6 Profils

6.1 Vue d'ensemble

L'Article 6 comporte une liste des catégories qu'un *Profil* peut regrouper, une liste des *Profils* désignés et la liste détaillée de chaque *Profil,* y compris les *Unités de Conformité* définies de manière directe et les sous-*Profils* inclus dans le *Profil*.

6.2 Liste des profils

Le Tableau 22 répertorie les *Profils*. Le tableau des *Profils* est ordonné par catégorie de *Profil,* puis classé dans l'ordre alphabétique du 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. Cet URI figure également sur le *Certificat de logiciel* associé au *Profil*. L'URI est sensible à la casse.

Une application (*Client* ou *Serveur*) doit mettre en œuvre toutes les *Unités de Conformité* dans un *Profil* afin d'être conforme à ce dernier. Certains *Profils* comportent des *Unités de Conformité* facultatives. Une *Unité de Conformité* facultative signifie qu'une application a la possibilité de ne pas prendre en charge l'*Unité de Conformité*. Toutefois, si elle est prise en charge, l'application doit satisfaire à tous les essais associés à l'*Unité de Conformité*. Par exemple, certaines *Unités de Conformité* exigent la disponibilité d'éléments de modèle d'informations spécifiques. Elles sont, par conséquent, définies comme facultatives de manière à pouvoir ignorer les éléments du modèle d'informations. Si un *Serveur* souhaite être classé comme prenant en charge l'*Unité de Conformité* facultative, il doit alors inclure tous les éléments de modèle d'informations requis dans la configuration fournie pour les essais de certification. La prise en charge des *Unités de Conformité* facultatives est décrite dans le certificat produit par les essais associés. Les *Unités de Conformité* facultatives sont clairement identifiées dans le présent document comme partie intégrante du *Certificat de Logiciel* décrivant les *Profils* pris en charge par un produit. Le *Certificat de Logiciel* doit présenter toutes les *Unités de Conformité* facultatives et spécifier si elles sont prises en charge. Les affichages en ligne qui établissent la liste des *Profils* pris en charge par un produit doivent également inclure les *Unités de Conformité* facultatives. Certaines *Unités de Conformité* comprennent également les listes des Types de Données pris en charge ou des Sous-types facultatifs pris en charge, les listes sont gérées de la même façon que les *Unités de Conformité* facultatives. Toutes les exigences de consignation pour les *Unités de Conformité* facultatives s'appliquent également à ces listes de Types de Données ou de sous-types pris en charge.

Tableau 22 – Liste des profils

Profil	Catégorie associée	URI
Core Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/CoreFacet
Base Server Behaviour Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/Behaviour
Attribute WriteMask Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/AttributeWriteMask
File Access Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/FileAccess
Documentation – Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/Documentation
Embedded DataChange Subscription Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/EmbeddedDataChangeSubscription
Standard DataChange Subscription Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/StandardDataChangeSubscription
Enhanced DataChange Subscription Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/EnhancedDataChangeSubscription
Data Access Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/DataAccess
ComplexType Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/ComplexTypes
Standard Event Subscription Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/StandardEventSubscription
Address Space Notifier Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/AddressSpaceNotifier
A & C Base Condition Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/ACBaseCondition
A & C Address Space Instance Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/ACAddressSpaceInstance
A & C Enable Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/ACEnable
A & C Alarm Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/ACAlarm
A & C Acknowledgeable Alarm Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/ACAckAlarm
A & C Exclusive Alarming Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/ACExclusiveAlarming
A & C Non-Exclusive Alarming Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/ACNon-ExclusiveAlarming
A & C Previous Instances Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/ACPreviousInstances
A & C Dialog Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/ACDialog
A & E Wrapper Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/AEWrapper
Method Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/Methods
Auditing Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/Auditing
Node Management Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/NodeManagement
Client Redundancy Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/ClientRedundancy
Redundancy Transparent Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/TransparentRedundancy

Profil	Catégorie associée	URI
Redundancy Visible Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/VisibleRedundancy
Historical Raw Data Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/HistoricalRawData
Historical Aggregate Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/AggregateHistorical
Historical Access Structured Data Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/HistoricalStructuredData
Historical Data AtTime Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/HistoricalDataAtTime
Historical Access Modified Data Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/HistoricalModifiedData
Historical Annotation Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/HistoricalAnnotation
Historical Data Update Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/HistoricalDataUpdate
Historical Data Replace Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/HistoricalDataReplace
Historical Data Insert Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/HistoricalDataInsert
Historical Data Delete Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/HistoricalDataDelete
Base Historical Event Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/BaseHistoricalEvent
Historical Event Update Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/HistoricalEventUpdate
Historical Event Replace Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/HistoricalEventReplace
Historical Event Insert Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/HistoricalEventInsert
Historical Event Delete Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/HistoricalEventDelete
Aggregate Subscription Server Facet	Serveur	http://opcfoundation.org/UA-Profile/Server/AggregateSubscription
Nano Embedded Device Server Profile	Serveur	http://opcfoundation.org/UA-Profile/Server/NanoEmbeddedDevice
Micro Embedded Device Server Profile	Serveur	http://opcfoundation.org/UA-Profile/Server/MicroEmbeddedDevice
Embedded UA Server Profile	Serveur	http://opcfoundation.org/UA-Profile/Server/EmbeddedUA
Standard UA Server Profile	Serveur	http://opcfoundation.org/UA-Profile/Server/StandardUA
Core Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/Core
Base Client Behaviour Facet	Client	http://opcfoundation.org/UA-Profile/Client/Behaviour
Discovery Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/Discovery
AddressSpace Lookup Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/AddressSpaceLookup
Entry-Level SupportClient Facet	Client	http://opcfoundation.org/UA-Profile/Client/Entry-LevelSupport
Multi-Server Client Connection Facet	Client	http://opcfoundation.org/UA-Profile/Client/MultiServer
File Access Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/FileAccess
Documentation – Client	Client	http://opcfoundation.org/UA-Profile/Client/Documentation

Profil	Catégorie associée	URI
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
DataAccess Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/DataAccess
Event Subscriber Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/EventSubscriber
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 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 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
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 & 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
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 Annotation Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalAnnotation
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 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

Profil	Catégorie associée	URI
Historical Data Insert Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalInsertData
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 Access Modified Data Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalAccessModifiedData
Historical Structured Data AtTime Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalAtTimeStructuredData
Historical Structured Data Access Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalAccessStructuredData
Historical Structured Data Modified Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalModifiedStructuredData
Historical Structured Data Delete Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalDeleteStructuredData
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 Insert Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalInsertStructuredData
Historical Events Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalEvents
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
Historical Event Insert Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/HistoricalInsertEvents
Aggregate Subscriber Client Facet	Client	http://opcfoundation.org/UA-Profile/Client/AggregateSubscriber
User Token – Anonymous Facet	Sécurité	http://opcfoundation.org/UA-Profile/Security/UserToken/Anonymous
User Token – User Name Password Server Facet	Serveur, Sécurité	http://opcfoundation.org/UA-Profile/Security/UserToken-Server/UserNamePassword
User Token – X509 Certificate Server Facet	Serveur, Sécurité	http://opcfoundation.org/UA-Profile/Security/UserToken-Server/X509Certificate
User Token – Issued Token Server Facet	Serveur, Sécurité	http://opcfoundation.org/UA-Profile/Security/UserToken-Server/IssuedToken
User Token – Issued Token Windows Server Facet	Serveur, Sécurité	http://opcfoundation.org/UA-Profile/Security/UserToken-Server/IssuedTokenWindows
User Token – User Name Password Client Facet	Client, Sécurité	http://opcfoundation.org/UA-Profile/Security/UserToken-Client/UserNamePassword

Profil	Catégorie associée	URI
User Token – X509 Certificate Client Facet	Client, Sécurité	http://opcfoundation.org/UA-Profile/Security/UserToken-Client/X509Certificate
User Token – Issued Token Client Facet	Client, Sécurité	http://opcfoundation.org/UA-Profile/Security/UserToken-Client/IssuedToken
User Token – Issued Token Windows Client Facet	Client, Sécurité	http://opcfoundation.org/UA-Profile/Security/UserToken-Client/IssuedTokenWindows
UA-TCP UA-SC UA Binary	Transport	http://opcfoundation.org/UA-Profile/Transport/uatcp-uasc-uabinary
SOAP-HTTP WS-SC UA XML	Transport	http://opcfoundation.org/UA-Profile/Transport/soaphttp-wssc-uaxml
SOAP-HTTP WS-SC UA Binary	Transport	http://opcfoundation.org/UA-Profile/Transport/soaphttp-wssc-uabinary
SOAP-HTTP WS-SC UA XML-UA Binary	Transport	http://opcfoundation.org/UA-Profile/Transport/soaphttp-wssc-uaxml-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
Security User Access Control Full	Sécurité, Serveur	http://opcfoundation.org/UA-Profile/Security/UserAccessFull
Security User Access Control Base	Sécurité, Serveur	http://opcfoundation.org/UA-Profile/Security/UserAccessBase
Security Time Synchronization	Sécurité	http://opcfoundation.org/UA-Profile/Security/TimeSync
Best Practice – Audit Events	Sécurité, Serveur	http://opcfoundation.org/UA-Profile/Security/BestPracticeAuditEvents
Best Practice – Alarm Handling	Sécurité, Serveur	http://opcfoundation.org/UA-Profile/Security/BestPracticeAlarmHandling
Best Practice – Program Access	Sécurité, Serveur	http://opcfoundation.org/UA-Profile/Security/BestPracticeProgramAccess
Best Practice – Random Numbers	Sécurité	http://opcfoundation.org/UA-Profile/Security/BestPracticeRandomNumbers
Best Practice – Timeouts	Sécurité	http://opcfoundation.org/UA-Profile/Security/BestPracticeTimeouts
Best Practice – Administrative Access	Sécurité	http://opcfoundation.org/UA-Profile/Security/BestPracticeAdministrativeAccess
Best Practice – Strict Message Handling	Sécurité, Serveur	http://opcfoundation.org/UA-Profile/Security/BestPracticeStrictMessage
Best Practice – Alarm Handling Client	Client, Sécurité	http://opcfoundation.org/UA-Profile/Security/BestPracticeAlarmHandlingClient
Best Practice – Audit Events Client	Client, Sécurité	http://opcfoundation.org/UA-Profile/Security/BestPracticeAuditEventsClient
SecurityPolicy – None	Sécurité	http://opcfoundation.org/UA/SecurityPolicy#None
SecurityPolicy – Basic128Rsa15	Sécurité	http://opcfoundation.org/UA/SecurityPolicy#Basic128Rsa15
SecurityPolicy – Basic256	Sécurité	http://opcfoundation.org/UA/SecurityPolicy#Basic256

Profil	Catégorie associée	URI
SecurityPolicy – Basic256Sha256	Sécurité	http://opcfoundation.org/UA/SecurityPolicy#Basic256Sha256
TransportSecurity – TLS 1.0	Sécurité	http://opcfoundation.org/UA-Profile/TransportSecurity/TLS-1-0
TransportSecurity – TLS 1.1	Sécurité	http://opcfoundation.org/UA-Profile/TransportSecurity/TLS-1-1
TransportSecurity – TLS 1.2	Sécurité	http://opcfoundation.org/UA-Profile/TransportSecurity/TLS-1-2

Le contenu de chacun des *Profils* énumérés est décrit sous forme de tableau dans une section séparée. Chaque tableau peut contenir des références aux *Profils* et/ou *Unités de Conformité* supplémentaires. Le référencement à un *Profil* signifie son inclusion pleine et entière. Les *Unités de Conformité* sont référencées grâce à leur nom et leur groupe de conformité. Pour les détails des *Unités de Conformité*, il convient que le lecteur examine ces derniers dans la section de groupe de conformité appropriée.

6.3 Conventions applicables aux définitions des profils

Les conventions d'affectation des noms suivantes sont associées aux *Profils*:

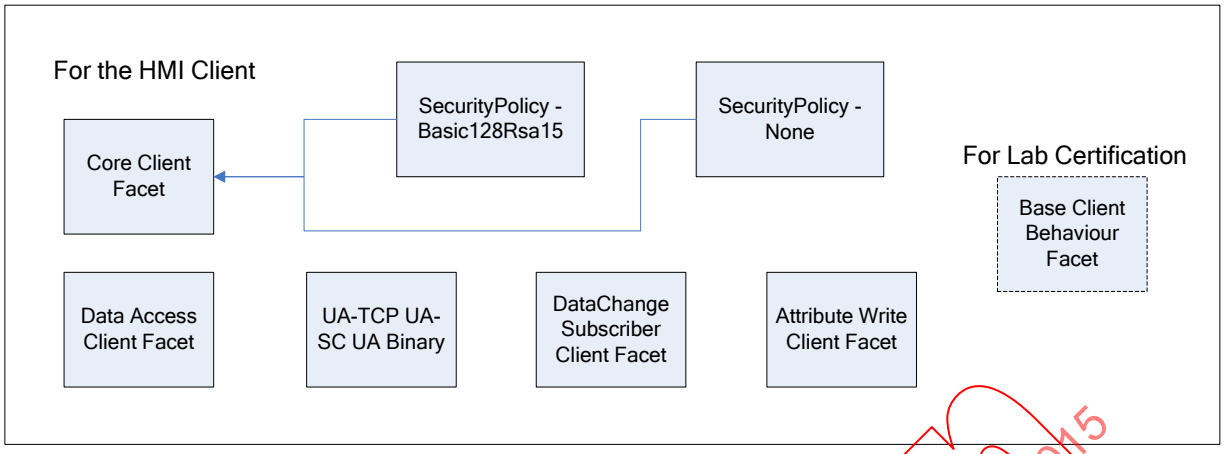
- Les *Profils* destinés aux *Serveurs* OPC UA comportent le terme *Serveur* dans leurs titres,
- Les *Profils* destinés aux *Clients* OPC UA comportent le terme *Client* dans leurs titres,
- Le terme Facet (Facette) dans le titre d'un *Profil* indique que le *Profil* considéré est supposé faire partie intégrante d'un autre *Profil* plus important ou concerne un aspect spécifique d'OPC UA. Les *Profils* dont le titre comporte le terme Facet (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 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, sélectionner les *Profils* qui incluent la fonctionnalité requise 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» ("Core *Client Facet*"), le *Profil* «Binaire UA UA-TCP UA-SC» ("UA-TCP UA-SC UA Binary"), la «*Facette Client* Accès aux Données» («Data Access *Client Facet*"), la «*Facette Client* Abonné aux Modifications de Données» ("DataChange Subscriber *Client* Facet") et la «*Facette Client Attribut* Écriture» ("*Attribute* Write *Client Facet*"). Si le *Client* est à soumettre à un essai par un laboratoire d'essai, il prend alors également en charge le *Profil* «*Client* Comportement de base» ("Base *Client* Behaviour"). La liste des *Profils* permet au *Client* de communiquer avec un *Serveur* OPC UA en utilisant le *Profil* UA-TCP/UA Sécurité/UA binaire (UA-TCP/UA Security/UA binary). 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 ligne directrice pratique de comportement.

La Figure 2 illustre la hiérarchie de *Profils* que cette application peut contenir. Cette figure est simplement une illustration, les *Profils* représentés pouvant varier.

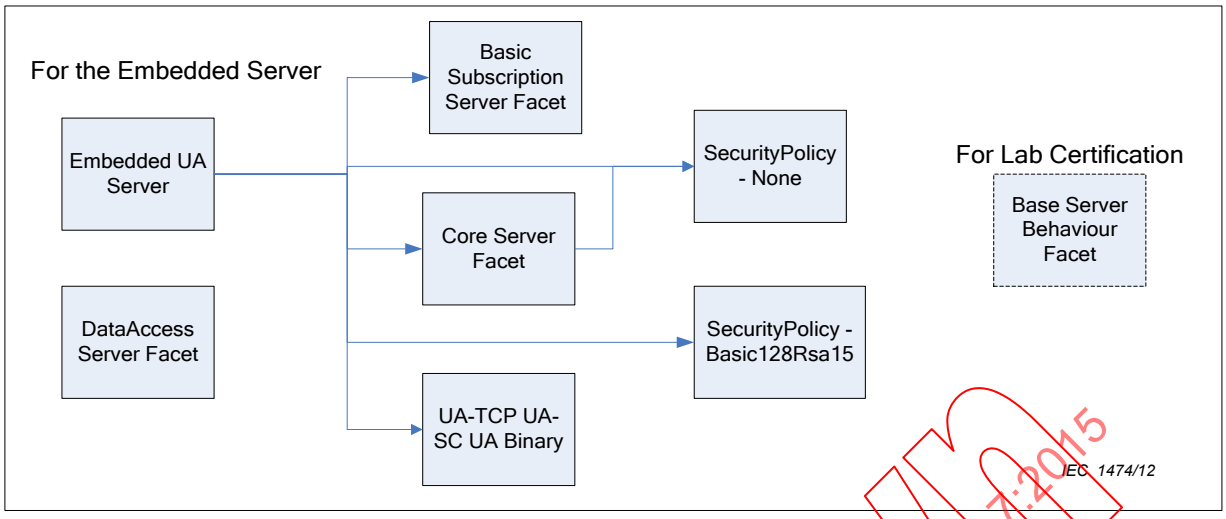


Légende

Anglais	Français
For the HMI Client	Pour l'interface IHM Client
Core Client Facet	Facette Client Principal
Data Access Client Facet	Facette Client Accès aux Données
SecurityPolicy Basic128Rsa15	Politique de Sécurité – Politique de Base 128Rsa15
UA-TCP UA-SC UA Binary	Profil Binaire UA UA-TCP UA-SC
DataChange Subscriber Client Facet	Facette Client Abonné aux Modifications de Données
SecurityPolicy None	Politique de Sécurité – Aucune
Attribute Write Client Facet	Facette Client Attribut Ecriture
For Lab Certification	Pour Certification en Laboratoire
Base Client Behaviour Facet	Facette Client Comportement de Base

Figure 2 – Echantillon IHM Client

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» (''Embedded UA Server'') et le *Profil* «Facette *Serveur* Accès aux données» (''DataAcess *Server* Facet''). Ce dispositif serait un dispositif à ressources limitées qui prendrait en charge les profils UA-TCP, UA-Sécurité, codage binaire UA, les abonnements aux données et le modèle de données DA. Il peut ne pas prendre en charge l'attribut facultatif «Écriture». La Figure 3 illustre la hiérarchie que cette application peut contenir: Cette figure est simplement une illustration, les *Profils* représentés pouvant varier.

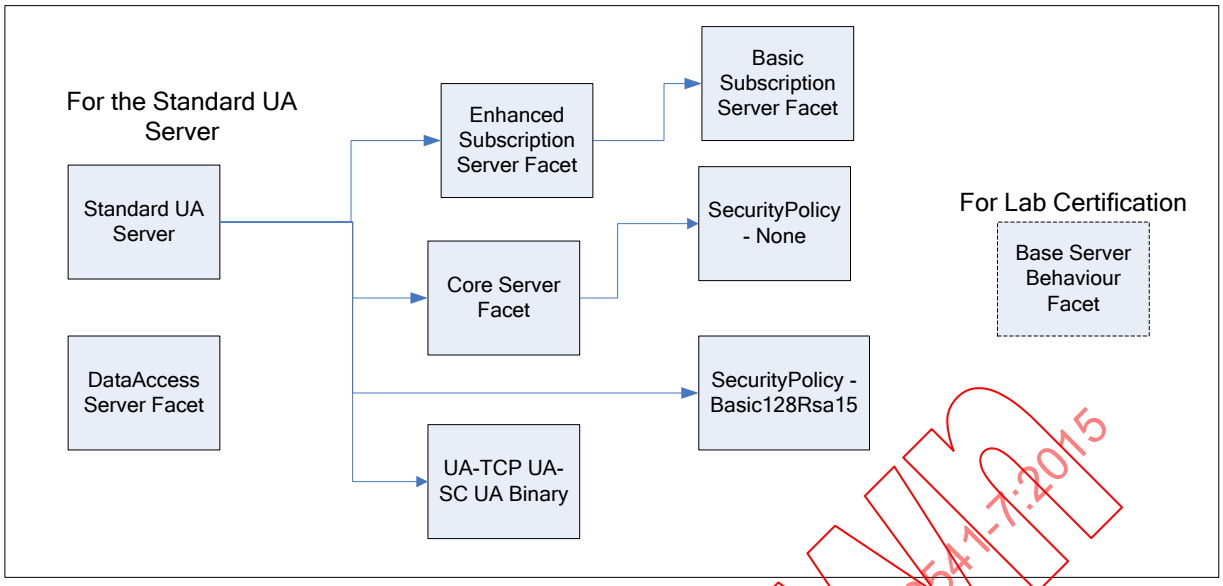


Légende

Anglais	Français
For the Embedded Server	Pour le Serveur Intégré
Embedded UA Server	Serveur UA Intégré
DataAccess Server Facet	Facette Serveur Accès aux Données
SecurityPolicy Basic128Rsa15	Politique de Sécurité – Politique de Base 128Rsa15
Basic Subscription Server Facet	Facette Serveur Abonnement de Base
Core Server Facet	Facette Serveur Principal
UA-TCP UA-SC UA Binary	Profil Binaire UA UA-TCP UA-SC
SecurityPolicy – None	Politique de Sécurité – Aucune
For Lab Certification	Pour Certification en Laboratoire
Base Server Behaviour Facet	Facette Comportement Serveur de Base

Figure 3 – Échantillon de Serveur intégré

Une autre application *Serveur* système simple peut choisir de prendre en charge: les *Profils* «*Serveur* UA normalisé» (“Standard UA *Serveur*”) et «Facette *Serveur* Accès aux données» (“DataAccess *Server* Facet”). Si le *Serveur* est à soumettre à un essai par un laboratoire d’essai, il prend alors également en charge le *Profil* «Comportement *Serveur* de base» (“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 illustre la hiérarchie que cette application peut contenir: il s’agit simplement d’une illustration, le *Profil* représenté pouvant varier.



Légende

Anglais	Français
For the Standard UA Server	Pour le Serveur UA Normalisé
Standard UA Server	Serveur UA Normalisé
DataAccess Server Facet	Facette Serveur Accès aux Données
SecurityPolicy Basic128Rsa15	Politique de Sécurité – Politique de Base 128Rsa15
Enhanced Subscription Server Facet	Facette Serveur Abonnement Amélioré
Core Server Facet	Facette Serveur Principal
UA-TCP UA-SC UA Binary	Profil Binaire UA UA-TCP UA-SC
SecurityPolicy – None	Politique de Sécurité – Aucune
For Lab Certification	Pour Certification en Laboratoire
Base Server Behaviour Facet	Facette Comportement Serveur de Base
Basic Subscription Server Facet	Facette Serveur Abonnement de Base

Figure 4 – Échantillon de Serveur UA normalisé

Si l'interface IHM *Client* présentée en exemple était reliée à l'un ou l'autre des *Serveurs* présentés en exemple, il est possible qu'elle ait à 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 est également possible qu'elle ait à 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, elle présenterait de plus grandes limites concernant les éléments relatifs à la performance, et elle permettrait les opérations d'écriture.

6.5 Tableaux des Profils

6.5.1 Introduction

Tous les paragraphes de 6.5 qui commencent par 6.5.2 décrivent les *Profils* sous forme de tableaux.

Chaque tableau comporte trois colonnes. La première colonne est une description du groupe de conformité dont fait partie intégrante *l'Unité de Conformité*. Ceci permet au lecteur de

déterminer facilement *l'Unité de Conformité*. Cette colonne peut également indiquer le «*Profil*» dans le cas où l'élément énuméré n'est pas une *Unité de Conformité*, mais un *Profil* inclus. La deuxième colonne donne une description succincte de *l'Unité de Conformité* ou du *Profil* inclus. La dernière colonne indique le caractère facultatif ou obligatoire de *l'Unité de Conformité*.

6.5.2 Facette Serveur principal (Core Server Facet)

Le Tableau 23 décrit les détails de la Facette *Serveur* principal. Cette Facette définit la fonctionnalité principale requise pour toute mise en œuvre d'un *Serveur* UA. La fonctionnalité principale comporte la capacité à découvrir des points d'extrémité, établir des canaux de communication sécurisés, créer des sessions, explorer l'*Espace d'adresses* 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 des Capacités *Objet* de ce dernier, ainsi que de tous les *Attributs* obligatoires pour les *Nœuds* dans l'*Espace d'adresses*, et l'Authentification avec le nom et le mot de passe de l'utilisateur. La prise en charge d'un type de système n'est pas requise, de même qu'il n'est pas nécessaire que le *Serveur* prenne en charge le cryptage et la signature des jetons d'identité de l'utilisateur (ceci suppose que le *Serveur* prend également en charge un transport sécurisé). Cette Facette a été étendue avec les *Unités de conformité* des Informations de base supplémentaires. Celles-ci sont facultatives pour la rétrocompatibilité. À l'avenir, l'*Unité de Conformité* "Info de base – Capacités du Serveur" sera requise, il est donc fortement recommandé que tous les *Serveurs* la prennent en charge. Pour une applicabilité générale, il est recommandé que les *Serveurs* prennent en charge plusieurs *Profils* de transport et de sécurité.

Tableau 23 – Facette Serveur principal

Groupe	Titre Unité de Conformité / *Profil*	Facultatif
Profil	Politique de sécurité – Aucune	Faux
Profil	Jeton d'Utilisateur – Facette Serveur Nom d'Utilisateur Mot de Passe	Faux
Modèle de l'Espace d'adresses	Base Espace d'adresses	Faux
Services Attributs	Attribut Lecture	Faux
Services Attributs	Attribut Indice d'écriture	Vrai
Services Attributs	Attribut Valeurs d'écriture	Vrai
Informations de base	Structure principale des informations de base	Faux
Informations de Base	OptionSet des informations de base	Vrai
Informations de Base	Règles de Modélisation Paramètre Fictif des informations de base	Vrai
Informations de Base	Capacités de Serveur des informations de base	Vrai
Informations de Base	ValueAsText des informations de base	Vrai
Services Découverte	Trouver Serveurs de découverte pour usage individuel	Faux
Services Découverte	Obtenir les points d'extrémité de découverte	Faux
Sécurité	Sécurité – Pas d'Authentification d'Application	Vrai
Sécurité	Administration Sécurité	Vrai
Services Session	Session de Base	Faux
Services Session	Session Comportement de service général	Faux
Services Session	Session Minimum 1	Faux
Services Vue	Vue de base	Faux
Services Vue	Vue Point de continuation minimum 01	Faux
Services Vue	Vue Enregistrement des nœuds (RegisterNodes)	Faux
Services Vue	Vue Traduire chemin de navigation (TranslateBrowsePath)	Faux

6.5.3 Facette Comportement Serveur de base (Base Server Behaviour Facet)

Le Tableau 24 décrit les détails de la Facette Comportement *Serveur* de base. 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, de définir le niveau de sécurité et de configurer le *Serveur de découverte*, ainsi que de spécifier le support sur lequel ce *Serveur* doit être enregistré.

Tableau 24 – Facette Comportement Serveur de base

Groupe	Titre Unité de Conformité / *Profil*	Facultatif
Services Découverte	Configuration de découverte	Faux
Protocole et Codage	Configuration de protocole	Faux
Sécurité	Administration Sécurité	Faux
Sécurité	Administration Sécurité – Schéma XML	Faux
Sécurité	Administration Certificat de sécurité	Faux

6.5.4 Facette *Serveur* Attribut WriteMask (Attribute WriteMask Server Facet)

Le Tableau 25 décrit les détails de la Facette *Serveur* Attribut WriteMask. Cette Facette définit la capacité à mettre à jour les caractéristiques des *Nœuds* individuels dans l'*Espace d'adresses* 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'*Espace d'adresses* et la limitation réelle des droits d'accès, comme décrit.

Tableau 25 – Facette *Serveur* Attribut WriteMask

Groupe	Titre Unité de Conformité / *Profil*	Facultatif
Profil	Contrôle Accès Utilisateur Sécurité de Base	Faux
Modèle d'Espace d'adresses	Espace d'adresses – UserWriteMask	Faux
Modèle d'Espace d'adresses	Espace d'adresses – UserWriteMask Multiniveaux	Vrai
Modèle d'Espace d'adresses	Espace d'adresses – WriteMask	Faux

6.5.5 Facette Serveur Accès Fichier (File Access Server Facet)

Le Tableau 26 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 26 – Facette Serveur Accès Fichier

Groupe	Titre Unité de Conformité / *Profil*	Facultatif
Informations de Base	Base Type de Fichier des Informations de Base	Faux
Informations de Base	Écriture Type de Fichier des Informations de Base	Vrai

6.5.6 Facette *Serveur* Documentation (Documentation Server Facet)

Le Tableau 27 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 27 – Facette Serveur Documentation

Groupe	Titre Unité de Conformité / *Profil*	Facultatif
Divers	Documentation – Installation	Faux
Divers	Documentation – Langages Multiple	Vrai
Divers	Documentation – En ligne	Vrai
Divers	Documentation – *Profils* pris en charge	Vrai
Divers	Documentation – Guide de Recherche de pannes	Vrai
Divers	Documentation – Guide Utilisateur	Faux