IEC TR 62541-1

Edition 2.0 2016-10

TECHNICAL REPORT

colour inside

OPC unified architecture – Part 1: Overview and concepts

IEC TR 62541-1:2016-10(en)

International Electrotechnical Commission

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 It also gives information on projects, replaced and withdrawn publications.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 20 000 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
65 000 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.

IEC TR 62541-1

Edition 2.0 2016-10

TECHNICAL REPORT

colour inside

OPC unified architecture –
Part 1: Overview and concepts

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

ICS 25.040.40; 35.100.01 ISBN 978-2-8322-3640-6

Warning! Make sure that you obtained this publication from an authorized distributor.

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

____

OPC UNIFIED ARCHITECTURE –

Part 1: Overview and concepts

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.

The main task of IEC technical committees is to prepare International Standards. However, a technical committee may propose the publication of a technical report when it has collected data of a different kind from that which is normally published as an International Standard, for example "state of the art".

IEC TR 62541-1, which is a technical report, has been prepared by subcommittee 65E: Devices and integration in enterprise systems, of IEC technical committee 65: Industrial-process measurement, control and automation.

The text of this technical report is based on the following documents:

Enquiry draft	Report on voting
65E/414/DTR	65E/463/RVC

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

This second edition cancels and replaces the first edition of IEC TR 62541-1, published in 2010.

This edition includes no technical changes with respect to the previous edition but includes updates to reflect changes or additions in normative parts of IEC 62541.

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

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

- Italics are used to denote a defined term or definition that appears in the “Terms and definition” clause in one of the parts of the series.
- Italics are also used to denote the name of a service input or output parameter or the name of a structure or element of a structure that are usually defined in tables.
- The italicized terms and names are also often written in camel-case (the practice of writing compound words or phrases in which the elements are joined without spaces, with each element's initial letter capitalized within the compound). For example the defined term is AddressSpace instead of Address Space. This makes it easier to understand that there is a single definition for AddressSpace, not separate definitions for Address and Space.

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

The committee has decided that the contents of this publication will remain unchanged until the stability date indicated on the IEC website 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.

A bilingual version of this publication may be issued at a later date.

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 1: Overview and concepts

1 Scope

This part of IEC 62541, which is a Technical Report, presents the concepts and overview of the OPC Unified Architecture (OPC UA). Reading this document is helpful to understand the remaining parts of this multi-part document set. Each of the other parts is briefly explained along with a suggested reading order.

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 62541-2, *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 62541-7, *OPC unified architecture – Part 7: Profiles*

IEC 62541-8, *OPC Unified Architecture – Part 8: Data Access*

IEC 62541-9, *OPC Unified Architecture – Part 9: Alarms and Conditions*

IEC 62541-10, *OPC Unified Architecture – Part 10: Programs*

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

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

3 Terms, definitions and abbreviations

3.1 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

3.1.1
AddressSpace
collection of information that an OPC UA *Server* makes visible to its *Clients*

Note 1 to entry: See IEC 62541-3 for a description of the contents and structure of the *Server* AddressSpace.

3.1.2
Aggregate
a function that calculates derived values from *Raw data*

Note 1 to entry: *Raw data* may be from a historian or buffered real time data. Common *Aggregates* include averages over a given time range, minimum over a time range and maximum over a time range.

3.1.3
Alarm
type of Event associated with a state condition that typically requires acknowledgement

Note 1 to entry: See IEC 62541-9 for a description of Alarms.

3.1.4
Attribute
primitive characteristic of a Node

Note 1 to entry: All Attributes are defined by OPC UA, and may not be defined by *Clients* or *Servers*. Attributes are the only elements in the AddressSpace permitted to have data values.

3.1.5
Certificate
digitally signed data structure that describes capabilities of a *Client* or *Server*

3.1.6
Client
software application that sends *Messages* to OPC UA *Servers* conforming to the *Services* specified in the IEC 62541 series of standards

3.1.7
Condition
generic term that is an extension to an *Event*

Note 1 to entry: A *Condition* represents the conditions of a system or one of its components and always exists in some state.

3.1.8
Communication Stack
layered set of software modules between the application and the hardware that provides various functions to encode, encrypt and format a *Message* for sending, and to decode, decrypt and unpack a *Message* that was received

3.1.9
Complex Data
data that is composed of elements of more than one primitive data type, such as a structure

3.1.10
Discovery
process by which OPC UA Client obtains information about OPC UA Servers, including endpoint and security information

3.1.11
Event
generic term used to describe an occurrence of some significance within a system or system component

3.1.12
EventNotifier
special *Attribute* of a *Node* that signifies that a *Client* may subscribe to that particular *Node* to receive *Notifications* of *Event* occurrences

3.1.13
Information Model
organizational framework that defines, characterizes and relates information resources of a given system or set of systems

Note 1 to entry: The core address space model supports the representation of Information Models in the AddressSpace. See IEC 62541-5 for a description of the base OPC UA Information Model.

3.1.14
Message
data unit conveyed between *Client* and *Server* that represents a specific *Service* request or response

3.1.15
Method
callable software function that is a component of an *Object*

3.1.16
MonitoredItem
Client-defined entity in the *Server* used to monitor Attributes or *EventNotifiers* for new values or Event occurrences and that generates Notifications for them

3.1.17
Node
fundamental component of an AddressSpace

3.1.18
NodeClass
class of a Node in an AddressSpace

Note 1 to entry: NodeClasses define the metadata for the components of the OPC UA Object Model. They also define constructs, such as Views, that are used to organize the AddressSpace.

3.1.19
Notification
generic term for data that announces the detection of an Event or of a changed Attribute value

Note 1 to entry: Notifications are sent in NotificationMessages.

3.1.20
NotificationMessage
Message published from a Subscription that contains one or more Notifications

3.1.21
Object
Node that represents a physical or abstract element of a system

Note 1 to entry: Objects are modelled using the OPC UA Object Model. Systems, subsystems and devices are examples of Objects. An Object may be defined as an instance of an ObjectType.

3.1.22
Object Instance
synonym for Object

Note 1 to entry: Not all Objects are defined by ObjectTypes.

3.1.23
ObjectType
Node that represents the type definition for an Object

3.1.24
Profile
specific set of capabilities to which a *Server* may claim conformance

Note 1 to entry: Each *Server* may claim conformance to more than one Profile.

Note 2 to entry: The set of capabilities are defined in IEC 62541-7.

3.1.25
Program
executable Object that, when invoked, immediately returns a response to indicate that execution has started, and then returns intermediate and final results through Subscriptions identified by the *Client* during invocation

3.1.26
Reference
explicit relationship (a named pointer) from one Node to another

Note 1 to entry: The Node that contains the Reference is the source Node, and the referenced Node is the target Node. All References are defined by ReferenceTypes.

3.1.27
ReferenceType
Node that represents the type definition of a Reference

Note 1 to entry: The ReferenceType specifies the semantics of a Reference. The name of a ReferenceType identifies how source Nodes are related to target Nodes and generally reflects an operation between the two, such as “A Contains B”.

3.1.28
RootNode
beginning or top Node of a *hierarchy*

Note 1 to entry: The RootNode of the OPC UA AddressSpace is defined in IEC 62541-5.

3.1.29
Server
software application that implements and exposes the Services specified in the IEC 62541 series of standards

3.1.30
Service
Client-callable operation in an OPC UA *Server*

Note 1 to entry: Services are defined in IEC 62541-4. A Service is similar to a method call in a programming language or an operation in a Web services WSDL contract.

3.1.31
Service Set
group of related *Services*

3.1.32
Session
logical long-running connection between a Client and a Server

Note 1 to entry: A Session maintains state information between Service calls from the Client to the Server.

3.1.33
Subscription
Client-defined endpoint in the Server, used to return Notifications to the Client

Note 1 to entry: ”Subscription” is a generic term that describes a set of Nodes selected by the Client (1) that the Server periodically monitors for the existence of some condition, and (2) for which the Server sends Notifications to the Client when the condition is detected.

3.1.34
Variable
Node that contains a value

3.1.35
View
specific subset of the AddressSpace that is of interest to the *Client*

3.2 Abbreviations

A&E	Alarms and Events
API	Application Programming Interface
COM	Component Object Model
DA	Data Access
DCS	Distributed Control System
DX	Data Exchange
HDA	Historical Data Access
HMI	Human-Machine Interface
LDAP	Lightweight Directory Access Protocol
MES	Manufacturing Execution System
OPC	OPC Foundation (a non-profit industry association) formerly an acronym for “OLE for Process Control”. No longer used anymore
PLC	Programmable Logic Controller
SCADA	Supervisory Control And Data Acquisition
SOAP	Simple Object Access Protocol
UA	Unified Architecture
UDDI	Universal Description, Discovery and Integration
UML	Unified Modelling Language
WSDL	Web Services Definition Language
XML	Extensible Mark-up Language

4 Structure of the OPC UA series

4.1 Specification organization

OPC UA is organized as a multi-part specification, as illustrated in Figure 1.

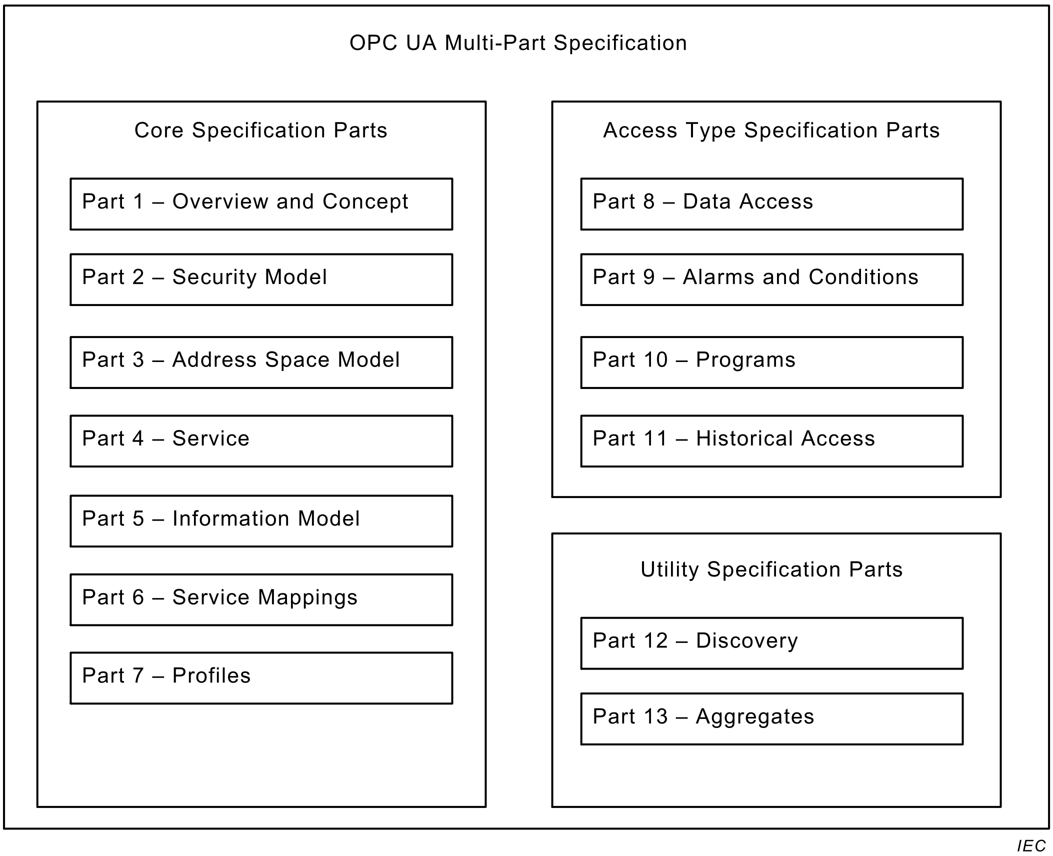

OPC UA Multi-Part Specification
Core Specification Parts
Part 1 – Overview and Concept
Part 2 – Security Model
Part 3 – Address Space Model
Part 4 – Service
Part 5 – Information Model
Part 6 – Service Mappings
Part 7 – Profiles
Access Type Specification Parts
Part 8 – Data Access
Part 9 – Alarms and Conditions
Part 10 – Programs
Part 11 – Historical Access
Utility Specification Parts
Part 12 – Discovery
Part 13 – Aggregates
IEC

Figure 1 – OPC UA Specification organization

IEC 62541-1 to IEC 62541-7 specify the core capabilities of OPC UA. These core capabilities define the structure of the OPC *AddressSpace* and the *Services* that operate on it. IEC 62541-8 to IEC 62541-11 apply these core capabilities to specific types of access previously addressed by separate OPC COM specifications, such as Data Access (DA), Alarms and Events (A&E) and Historical Data Access (HDA). IEC 62541-12 describes *Discovery* mechanisms for OPC UA and IEC 62541-13 describes ways of aggregating data.

Readers are encouraged to read IEC 62541-1 to IEC 62541-5 of the core specifications before reading IEC 62541-6 to IEC 62541-13. Some parts can be skipped depending on the needs of the reader. For example, a reader interested in Data Access should read IEC 62541-1 to IEC 62541-5 and then IEC 62541-8. References in IEC 62541-8 may direct the reader to other parts of this specification.

4.2 Core specification parts

IEC 62541-1 – Overview and concepts

Part 1 (this part) presents the concepts and overview of OPC UA.

IEC 62541-2 – Security Model

IEC 62541-2 describes the model for securing interactions between OPC UA *Clients* and OPC UA *Servers*.

IEC 62541-3 – Address Space Model

IEC 62541-3 describes the contents and structure of the *Server's AddressSpace*.

IEC 62541-4 – Services

IEC 62541-4 specifies the *Services* provided by OPC UA *Servers*.

IEC 62541-5 – Information Model

IEC 62541-5 specifies the types and their relationships defined for OPC UA *Servers*.

IEC 62541-6 – Mappings

IEC 62541-6 specifies the mappings to transport protocols and data encodings supported by OPC UA.

IEC 62541-7 – Profiles

IEC 62541-7 specifies the *Profiles* that are available for OPC *Clients* and *Servers*. These *Profiles* provide groups of *Services* or functionality that can be used for conformance level certification. *Servers* and *Clients* will be tested against the *Profiles*.

4.3 Access Type specification parts

IEC 62541-8 – Data Access

IEC 62541-8 specifies the use of OPC UA for data access.

IEC 62541-9 – Alarms and Conditions

IEC 62541-9 specifies use of OPC UA support for access to *Alarms* and *Conditions*. The base system includes support for simple *Events*; this specification extends that support to include support for *Alarms* and *Conditions*.

IEC 62541-10 – Programs

IEC 62541-10 specifies OPC UA support for access to *Programs*.

IEC 62541-11 – Historical Access

IEC 62541-11 specifies use of OPC UA for historical access. This access includes both historical data and historical *Events*.

4.4 Utility specification parts

IEC 62541-12 – Discovery

IEC 62541-12 specifies how *Discovery Servers* operate in different scenarios and describes how UA *Clients* and *Servers* should interact with them. It also defines how UA related information should be accessed using common directory service protocols such as UDDI and LDAP.

IEC 62541-13 – Aggregates

IEC 62541-13 specifies how to compute and return aggregates like minimum, maximum, average etc. Aggregates can be used with current and historical data.

5 IEC 62541 standards – Overview

5.1 UA scope

OPC UA is applicable to manufacturing software in application areas such as Field Devices, Control Systems, Manufacturing Execution Systems and Enterprise Resource Planning Systems. These systems are intended to exchange information and to use command and control for industrial processes. OPC UA defines a common infrastructure model to facilitate this information exchange. OPC UA specifies the following:

- the information model to represent structure, behaviour and semantics;
- the message model to interact between applications;
- the communication model to transfer the data between end-points;
- the conformance model to guarantee interoperability between systems.

5.2 General

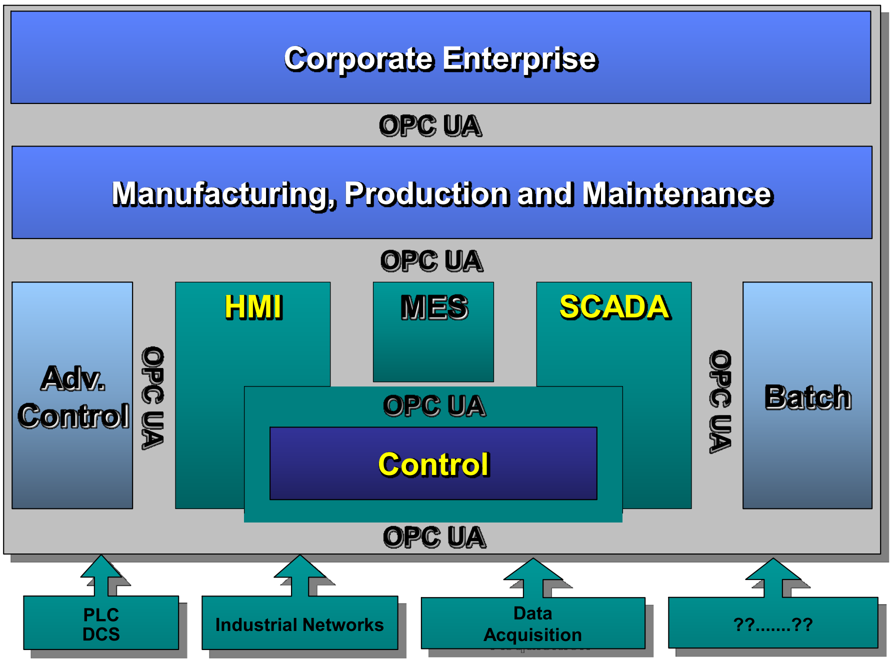
OPC UA is a platform-independent standard through which various kinds of systems and devices can communicate by sending *Messages* between *Clients* and *Servers* over various types of networks. It supports robust, secure communication that assures the identity of *Clients* and *Servers* and resists attacks. OPC UA defines sets of *Services* that *Servers* may provide, and individual *Servers* specify to *Clients* what *Service* sets they support. Information is conveyed using OPC UA-defined and vendor-defined data types, and *Servers* define object models that *Clients* can dynamically discover. *Servers* can provide access to both current and historical data, as well as *Alarms* and *Events* to notify *Clients* of important changes. OPC UA can be mapped onto a variety of communication protocols and data can be encoded in various ways to trade off portability and efficiency.

5.3 Design goals

OPC UA provides a consistent, integrated *AddressSpace* and service model. This allows a single OPC UA *Server* to integrate data, *Alarms* and *Events*, and history into its *AddressSpace*, and to provide access to them using an integrated set of *Services*. These *Services* also include an integrated security model.

OPC UA also allows *Servers* to provide *Clients* with type definitions for the *Objects* accessed from the *AddressSpace*. This allows information models to be used to describe the contents of the *AddressSpace*. OPC UA allows data to be exposed in many different formats, including binary structures and XML documents. The format of the data may be defined by OPC, other standard organizations or vendors. Through the *AddressSpace*, *Clients* can query the *Server* for the metadata that describes the format for the data. In many cases, *Clients* with no pre-programmed knowledge of the data formats will be able to determine the formats at runtime and properly utilize the data.

OPC UA adds support for many relationships between *Nodes* instead of being limited to just a single hierarchy. In this way, an OPC UA *Server* may present data in a variety of hierarchies tailored to the way a set of *Clients* would typically like to view the data. This flexibility, combined with support for type definitions, makes OPC UA applicable to a wide array of problem domains. As illustrated in Figure 2, OPC UA is not targeted at just the SCADA, PLC and DCS interface, but also as a way to provide greater interoperability between higher level functions.



IEC

Figure 2 – OPC UA Target applications

OPC UA is designed to provide robustness of published data. A major feature of all OPC servers is the ability to publish data and *Event Notifications*. OPC UA provides mechanisms for *Clients* to quickly detect and recover from communication failures associated with these transfers without having to wait for long timeouts provided by the underlying protocols.

OPC UA is designed to support a wide range of *Servers*, from plant floor PLCs to enterprise *Servers*. These *Servers* are characterized by a broad scope of size, performance, execution platforms and functional capabilities. Therefore, OPC UA defines a comprehensive set of capabilities, and *Servers* may implement a subset of these capabilities. To promote interoperability, OPC UA defines subsets, referred to as *Profiles*, to which *Servers* may claim conformance. *Clients* can then discover the *Profiles* of a *Server*, and tailor their interactions with that *Server* based on the *Profiles*. *Profiles* are defined in IEC 62541-7.

The OPC UA specifications are layered to isolate the core design from the underlying computing technology and network transport. This allows OPC UA to be mapped to future technologies as necessary, without negating the basic design. Mappings and data encodings are described in IEC 62541-6. Two data encodings are defined:

- XML/text,
- UA Binary.

In addition, three transport protocols are defined:

- OPC UA TCP,
- SOAP/HTTP,
- HTTPS.

Clients and Servers that support multiple transports and encodings will allow the end users to make decisions about tradeoffs between performance and XML Web service compatibility at the time of deployment, rather than having these tradeoffs determined by the OPC vendor at the time of product definition.

OPC UA is designed as the migration path for OPC clients and servers that are based on Microsoft COM technology. Care has been taken in the design of OPC-UA so that existing data exposed by OPC COM servers (DA, HDA and A&E) can easily be mapped and exposed via OPC UA. Vendors may choose to migrate their products natively to OPC UA or use external wrappers to convert from OPC COM to OPC UA and vice-versa. Each of the previous OPC specifications defined its own address space model and its own set of *Services*. OPC UA unifies the previous models into a single integrated address space with a single set of *Services*.

5.4 Integrated models and services

5.4.1 Security model

5.4.1.1 General

OPC UA security is concerned with the authentication of *Clients* and *Servers*, the authentication of users, the integrity and confidentiality of their communications, and the verifiability of claims of functionality. It does not specify the circumstances under which various security mechanisms are required. That specification is crucial, but it is made by the designers of the system at a given site and may be specified by other standards.

Rather, OPC UA provides a security model, described in IEC TR 62541-2, in which security measures can be selected and configured to meet the security needs of a given installation. This model includes security mechanisms and parameters. In some cases, the mechanism for exchanging security parameters is defined, but the way that applications use these parameters is not. This framework also defines a minimum set of security *Profiles* that all UA *Servers* support, even though they may not be used in all installations. Security *Profiles* are defined in IEC 62541-7.

5.4.1.2 *Discovery* and *Session* establishment

Application level security relies on a secure communication channel that is active for the duration of the application *Session* and ensures the integrity of all *Messages* that are exchanged. This means users need to be authenticated only once, when the application *Session* is established. The mechanisms for discovering OPC UA Servers and establishing secure communication channels and application *Sessions* are described in IEC 62541-4 and IEC 62541-6. Additional information about the *Discovery* process is described in IEC 62541-12.

When a *Session* is established, the *Client* and *Server* applications negotiate a secure communications channel. Digital (X.509) *Certificates* are utilized to identify the *Client* and *Server* and the capabilities that they provide. Authority-generated software *Certificates* indicate the OPC UA *Profiles* that the applications implement and the OPC UA certification level reached for each *Profile*1. The details of each *Profile* and the *Certificates* are specified in IEC 62541-7. *Certificates* issued by other organizations may also be exchanged during *Session* establishment.

The *Server* further authenticates the user and authorizes subsequent requests to access *Objects* in the *Server*. Authorization mechanisms, such as access control lists, are not specified by the OPC UA specification. They are application or system-specific.

5.4.1.3 Auditing

OPC UA includes support for security audit trails with traceability between *Client* and *Server* audit logs. If a security-related problem is detected at the *Server*, the associated *Client* audit log entry can be located and examined. OPC UA also provides the capability for *Servers* to generate *Event Notifications* that report auditable *Events* to *Clients* capable of processing and

[1] The OPC Foundation is an OPC UA Certificate authority.

logging them. OPC UA defines security audit parameters that can be included in audit log entries and in audit *Event Notifications*. IEC 62541-5 defines the data types for these parameters. Not all *Servers* and *Clients* provide all of the auditing features. *Profiles,* found in IEC 62541-7, indicate which features are supported.

5.4.1.4 Transport security

OPC UA security complements the security infrastructure provided by most web service capable platforms.

Transport level security can be used to encrypt and sign *Messages*. Encryption and signatures protect against disclosure of information and protect the integrity of *Messages*. Encryption capabilities are provided by the underlying communications technology used to exchange *Messages* between OPC UA applications. IEC 62541-7 defines the encryption and signature algorithms to be used for a given *Profile*.

5.4.2 Integrated *AddressSpace* model

The set of *Objects* and related information that the OPC UA *Server* makes available to *Clients* is referred to as its *AddressSpace*. The OPC UA *AddressSpace* represents its contents as a set of *Node*s connected by *References*.

Primitive characteristics of *Node*s are described by OPC-defined *Attributes*. *Attributes* are the only elements of a *Server* that have data values. Data types that define attribute values may be simple or complex.

*Node*s in the *AddressSpace* are typed according to their use and their meaning. *NodeClasses* define the metadata for the OPC UA *AddressSpace*. IEC 62541-3 defines the OPC UA *NodeClasses*.

The *Base NodeClass* defines *Attributes* common to all *Node*s, allowing identification, classification and naming. Each *NodeClass* inherits these *Attributes* and may additionally define its own *Attributes*.

To promote interoperability of *Clients* and *Servers*, the OPC UA *AddressSpace* is structured hierarchically with the top levels the same for all *Servers*. Although *Node*s in the *AddressSpace* are typically accessible via the hierarchy, they may have *References* to each other, allowing the *AddressSpace* to represent an interrelated network of *Node*s. The model of the *AddressSpace* is defined in IEC 62541-3.

OPC UA *Servers* may subset the *AddressSpace* into *Views* to simplify *Client* access. Subclause 6.3.4.3 describes *AddressSpace Views* in more detail.

5.4.3 Integrated object model

The OPC UA Object Model provides a consistent, integrated set of *NodeClasses* for representing *Objects* in the *AddressSpace*. This model represents *Objects* in terms of their *Variables, Events* and *Methods*, and their relationships with other *Objects*. IEC 62541-3 describes this model.

The OPC UA object model allows *Servers* to provide type definitions for *Objects* and their components. Type definitions may be subclassed. They also may be common or they may be system-specific. *ObjectTypes* may be defined by standards organizations, vendors or end-users.

This model allows data, *Alarms* and *Events*, and their history to be integrated into a single OPC UA *Server*. For example, OPC UA *Servers* are able to represent a temperature transmitter as an *Object* that is composed of a temperature value, a set of alarm parameters, and a corresponding set of alarm limits.

5.4.4 Integrated services

The interface between OPC UA *Clients* and *Servers* is defined as a set of *Services.* These *Services* are organized into logical groupings called *Service Sets*. *Service Sets* are discussed in Clause 7 and specified in IEC 62541-4.

OPC UA *Services* provide two capabilities to *Clients*. They allow *Clients* to issue requests to *Servers* and receive responses from them. They also allow *Clients* to subscribe to *Servers* for *Notifications*. *Notifications* are used by the *Server* to report occurrences such as *Alarms*, data value changes, *Events*, and *Program* execution results.

OPC UA *Messages* may be encoded as XML text or in binary format for efficiency purposes. They may be transferred using multiple underlying transports, for example TCP or web services over HTTP. *Servers* may provide different encodings and transports as defined by IEC 62541-6.

5.5 Sessions

OPC UA requires a stateful model. The state information is maintained inside an application *Session*. Examples of state-information are *Subscriptions*, user credentials and continuation points for operations that span multiple requests.

Sessions are defined as logical connections between *Clients* and *Servers*. *Servers* may limit the number of concurrent *Sessions* based on resource availability, licensing restrictions, or other constraints. Each *Session* is independent of the underlying communications protocols. Failures of these protocols do not automatically cause the *Session* to terminate. *Sessions* terminate based on *Client* or *Server* request, or based on inactivity of the *Client*. The inactivity time interval is negotiated during *Session* establishment.

5.6 Redundancy

The design of OPC UA ensures that vendors can create redundant *Clients* and redundant *Servers* in a consistent manner. Redundancy may be used for high availability, fault tolerance and load balancing. The details for redundancy are found in IEC 62541-4. Only some *Profiles* IEC 62541-7 will require redundancy support, but not the base *Profile*.

6 Systems concepts

6.1 Overview

The OPC UA systems architecture models OPC UA *Clients* and *Servers* as interacting partners. Each system may contain multiple *Clients* and *Servers*. Each *Client* may interact concurrently with one or more *Servers*, and each *Server* may interact concurrently with one or more *Clients*. An application may combine *Server* and *Client* components to allow interaction with other *Servers* and *Clients* as described in 6.3.7.

OPC UA Clients and Servers are described in the 6.2 and 6.3. Figure 3 illustrates the architecture that includes a combined *Server* and *Client*.

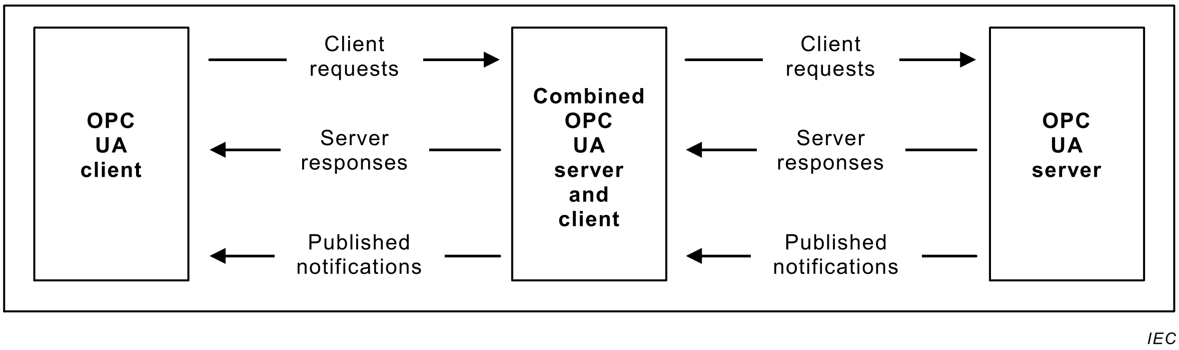


Figure 3 – OPC UA System architecture

6.2 OPC UA *Clients*

The OPC UA *Client* architecture models the *Client* endpoint of client/server interactions. Figure 4 illustrates the major elements of a typical OPC UA *Client* and how they relate to each other.

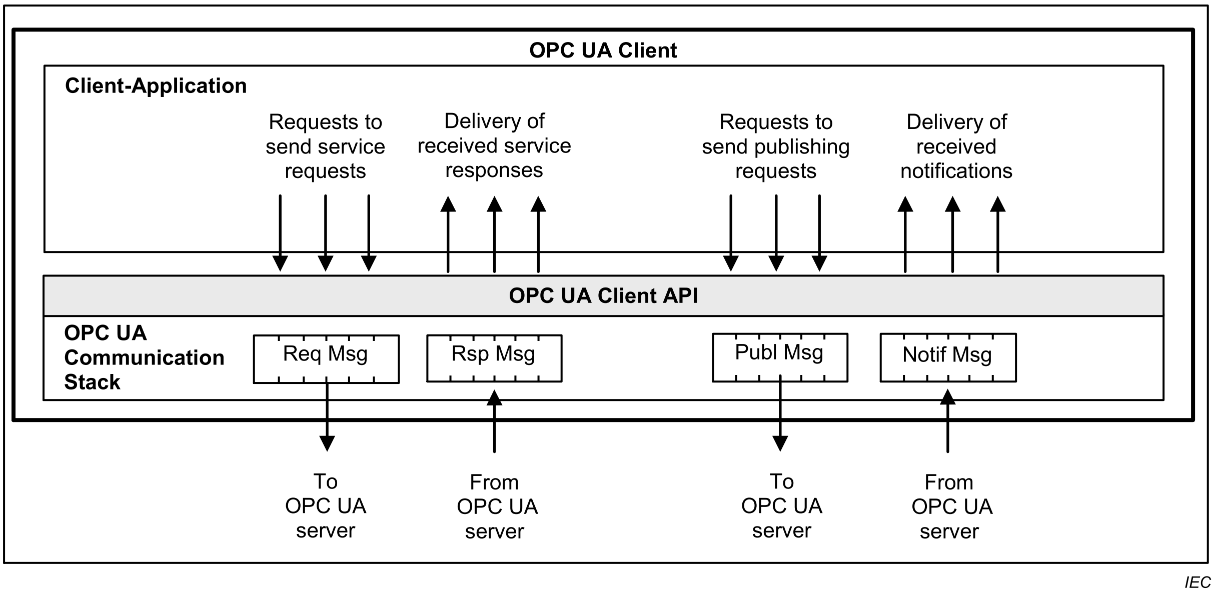

OPC UA Client
Client-Application
Requests to send service requests
Delivery of received service responses
Requests to send publishing requests
Delivery of received notifications
OPC UA Client API
OPC UA Communication Stack
Req Msg
Rsp Msg
Publ Msg
Notif Msg
To OPC UA server
From OPC UA server
To OPC UA server
From OPC UA server
IEC

Figure 4 – OPC UA Client architecture

The *Client* Application is the code that implements the function of the *Client*. It uses the OPC UA *Client* API to send and receive OPC UA *Service* requests and responses to the OPC UA *Server*. The *Services* defined for OPC UA are described in Clause 7, and specified in IEC 62541-4.

Note that the “OPC UA *Client* API” is an internal interface that isolates the *Client* application code from an OPC UA Communication Stack. The OPC UA Communication Stack converts OPC UA *Client* API calls into *Messages* and sends them through the underlying communications entity to the *Server* at the request of the *Client* application. The OPC UA Communication Stack also receives response and *NotificationMessages* from the underlying communications entity and delivers them to the *Client* application through the OPC UA *Client* API.

6.3 OPC UA *Servers*

6.3.1 General

The OPC UA *Server* architecture models the *Server* endpoint of client/server interactions. Figure 5 illustrates the major elements of the OPC UA *Server* and how they relate to each other.

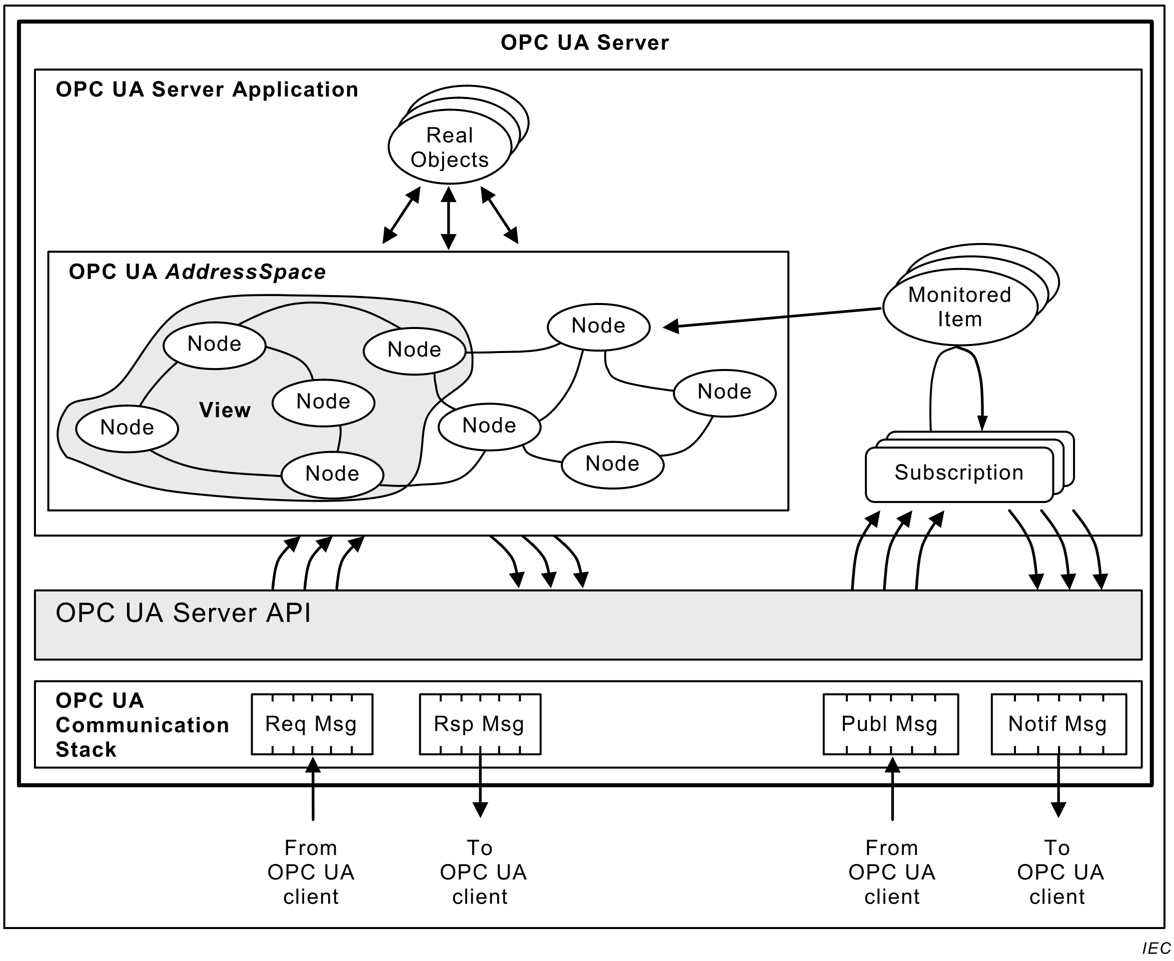


Figure 5 – OPC UA Server architecture

6.3.2 Real objects

Real objects are physical or software objects that are accessible by the OPC UA *Server* application or that it maintains internally. Examples include physical devices and diagnostics counters.

6.3.3 OPC UA *Server* application

The OPC UA *Server* application is the code that implements the function of the *Server*. It uses the OPC UA *Server* API to send and receive OPC UA *Messages* from OPC UA *Clients*. Note that the “OPC UA *Server* API” is an internal interface that isolates the *Server* application code from an OPC UA Communication Stack.

6.3.4 OPC UA AddressSpace

6.3.4.1 AddressSpace Nodes

The *AddressSpace* is modelled as a set of *Node*s accessible by *Clients* using OPC UA *Services* (interfaces and methods). *Node*s in the *AddressSpace* are used to represent real objects, their definitions and their *References* to each other.

6.3.4.2 *AddressSpace* organization

IEC 62541-3 contains the details of the meta model “building blocks” used to create an *AddressSpace* out of interconnected *Nodes* in a consistent manner. *Servers* are free to organize their *Node*s within the *AddressSpace* as they choose. The use of *References* between *Node*s permits *Servers* to organize the *AddressSpace* into hierarchies, a full mesh network of *Node*s, or any possible mix.

IEC 62541-5 defines OPC UA *Nodes* and *References* and their expected organization in the *AddressSpace*. Some *Profiles* will not require that all of the UA *Nodes* be implemented.

6.3.4.3 AddressSpace Views

A *View* is a subset of the *AddressSpace*. *Views* are used to restrict the *Node*s that the *Server* makes visible to the *Client*, thus restricting the size of the *AddressSpace* for the *Service* requests submitted by the *Client*. The default *View* is the entire *AddressSpace*. *Servers* may optionally define other *Views*. *Views* hide some of the *Node*s or *References* in the *AddressSpace*. *Views* are visible via the *AddressSpace* and *Clients* are able to browse *Views* to determine their structure. *Views* are often hierarchies, which are easier for *Clients* to navigate and represent in a tree.

6.3.4.4 Support for information models

The OPC UA *AddressSpace* supports information models. This support is provided through:

a) *Node References* that allow *Objects* in the *AddressSpace* to be related to each other,

b) *ObjectType Node*s that provide semantic information for real *Objects* (type definitions),

c) *ObjectType Node*s to support subclassing of type definitions,

d) Data type definitions exposed in the *AddressSpace* that allow industry specific data types to be used,

e) OPC UA companion standards that permit industry groups to define how their specific information models are to be represented in OPC UA *Server AddressSpace*s.

6.3.5 Publisher/subscriber entities

6.3.5.1 MonitoredItems

MonitoredItems are entities in the *Server* created by the *Client* that monitor *AddressSpace Node*s and their real-world counterparts. When they detect a data change or an event/alarm occurrence, they generate a *Notification* that is transferred to the *Client* by a *Subscription*.

6.3.5.2 Subscriptions

A *Subscription* is an endpoint in the *Server* that publishes *Notifications* to *Clients*. *Clients* control the rate at which publishing occurs by sending Publish *Messages*.

6.3.6 OPC UA *Service* Interface

6.3.6.1 General

The *Services* defined for OPC UA are described in Clause 7, and specified in IEC 62541-4.

6.3.6.2 Request/response *Services*

Request/response *Services* are *Services* invoked by the *Client* through the OPC UA *Service* Interface to perform a specific task on one or more *Node*s in the *AddressSpace* and to return a response.

6.3.6.3 Publisher *Services*

Publisher *Services* are *Services* invoked through the OPC UA *Service* Interface for the purpose of periodically sending *Notifications* to *Clients*. Notifications include *Events*, *Alarms*, data changes and *Program* outputs.

6.3.7 *Server* to *Server* interactions

Server to *Server* interactions are interactions in which one *Server* acts as a *Client* of another *Server*. *Server* to *Server* interactions allow for the development of servers that:

a) exchange information with each other on a peer-to-peer basis, this could include redundancy or remote *Servers* that are used for maintaining system wide type definitions(see Figure 6),
b) are chained in a layered architecture of *Servers* to provide:
   1) aggregation of data from lower-layer *Servers*,
   2) higher-layer data constructs to *Clients*, and
   3) concentrator interfaces to *Clients* for single points of access to multiple underlying *Servers*.

Figure 6 illustrates interactions between *Servers*.

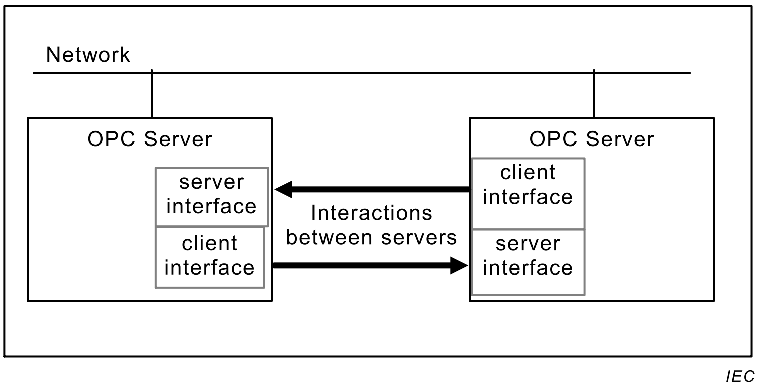


Figure 6 – Peer-to-peer interactions between Servers

Figure 7 extends the previous example and illustrates the chaining of OPC UA *Servers* together for vertical access to data in an enterprise.

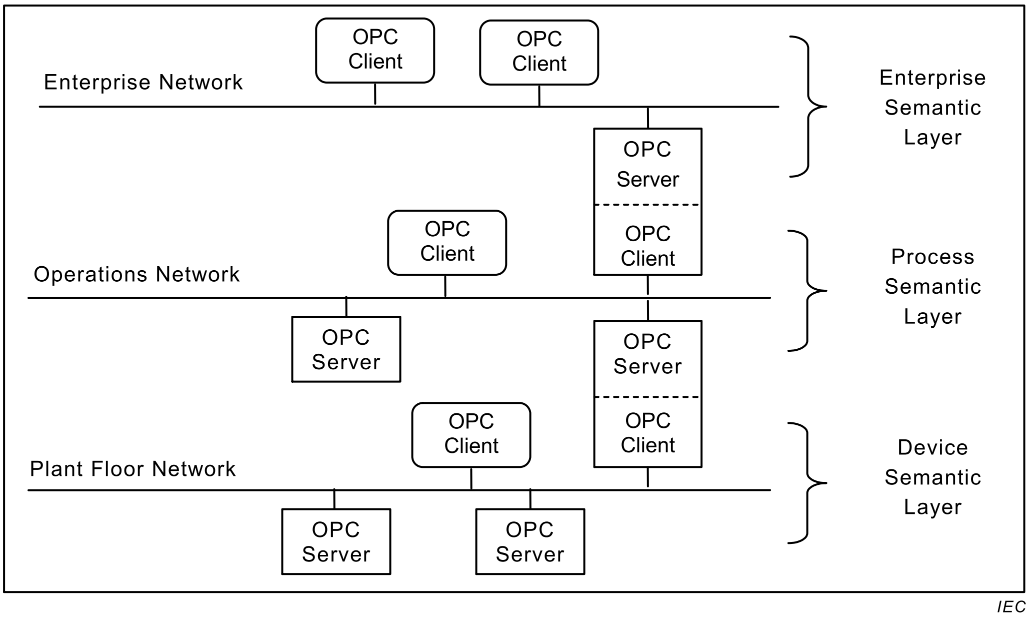


Figure 7 – Chained Server example

7 Service Sets

7.1 General

OPC UA *Services* are divided into *Service Sets*, each defining a logical grouping of *Services* used to access a particular aspect of the *Server*. The *Service Sets* are described below. The *Service Sets* and their *Services* are specified in IEC 62541-4. Whether or not a *Server* supports a *Service Set*, or a specific *Service* within a *Service Set,* is defined by its *Profile*. *Profiles* are described in IEC 62541-7.

7.2 Discovery Service Set

This *Service Set* defines *Services* used to discover OPC UA *Servers* that are available in a system. It also provides a manner in which clients can read the security configuration required for connection to the *Server*. The *Discovery Services* are implemented by individual *Servers* and by dedicated *Discovery Servers*. Well-known dedicated *Discovery Servers* provide a way for clients to discover all registered OPC UA *Servers*. IEC 62541-12 describes how to use the *Discovery Services* with dedicated *Discovery Servers.*

7.3 SecureChannel Service Set

This *Service Set* defines *Services* used to open a communication channel that ensures the confidentiality and integrity of all *Messages* exchanged with the *Server*. The base concepts for UA security are defined in IEC TR 62541-2.

The *SecureChannel Services* are unlike other *Services* because they are typically not implemented by the *UA application* directly. Instead, they are provided by the communication stack that the *UA application* is built on. For example, a UA *Server* may be built on a SOAP stack that allows applications to establish a *SecureChannel* using the WS-SecureConversation specification. In these cases, the *UA application* simply needs to verify that a WS-SecureConversation is active whenever it receives a *Message*. IEC 62541-6 describes how the *SecureChannel Services* are implemented with different types of communication stacks.

A *SecureChannel* is a long-running logical connection between a single *Client* and a single *Server*. This channel maintains a set of keys that are known only to the *Client* and *Server* and that are used to authenticate and encrypt *Messages* sent across the network. The *SecureChannel Services* allow the *Client* and *Server* to securely negotiate the keys to use.

The exact algorithms used to authenticate and encrypt *Messages* are described in the security policies for a *Server*. These policies are exposed via the *Discovery Service Set*. A *Client* selects the appropriate endpoint that supports the desired security policy by the *Server* when it creates a *SecureChannel*.

When a *Client* and *Server* are communicating via a *SecureChannel* they verify that all incoming *Messages* have been signed and/or encrypted according to the security policy. A UA application is expected to ignore any *Message* that does not conform to the security policy for the channel.

A *SecureChannel* is separate from the *UA Application Session*; however, a single *UA Application Session* may only be accessed via a single *SecureChannel*. This implies that the *UA application* is able to determine what *SecureChannel* is associated with each *Message*. A communication stack that provides a *SecureChannel* mechanism but that does not allow the application to know what *SecureChannel* was used for a given *Message* cannot be used to implement the *SecureChannel Service Set*.

The relationship between the *UA Application Session* and the *SecureChannel* is illustrated in Figure 8. The UA applications use the communication stack to exchange *Messages*. First, the *SecureChannel Services* are used to establish a *SecureChannel* between the two communication stacks, allowing them to exchange *Messages* in a secure way. Second, the UA applications use the *Session Service Set* to establish a *UA application Session*.

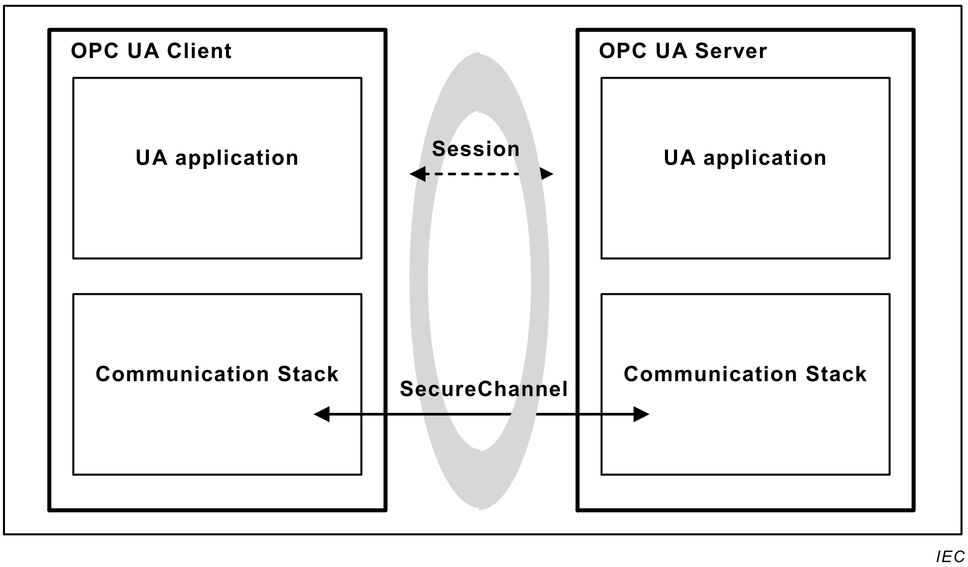


Figure 8 – SecureChannel and Session Services

7.4 Session Service Set

This *Service Set* defines *Services* used to establish an application-layer connection in the context of a *Session* on behalf of a specific user.

7.5 NodeManagement Service Set

The *NodeManagement Service Set* allows *Clients* to add, modify, and delete *Nodes* in the *AddressSpace*. These *Services* provide an interface for the configuration of *Servers*.

7.6 View Service Set

Views are publicly defined, *Server*-created subsets of the *AddressSpace*. The entire *AddressSpace* is the default *View*, and therefore, the *View Services* are capable of operating on the entire *AddressSpace*. Future versions of this specification may also define *Services* to create *Client* defined *Views*.

The *View Service Set* allows *Clients* to discover *Node*s in a *View* by browsing. Browsing allows *Clients* to navigate up and down the hierarchy, or to follow *References* between *Node*s contained in the *View*. In this manner, browsing also allows *Clients* to discover the structure of the *View*.

7.7 Query Service Set

The Query Service Set allows users to access the address space without browsing and without knowledge of the logical schema used for internal storage of the data.

Querying allows *Clients* to select a subset of the *Node*s in a *View* based on some *Client*-provided filter criteria. The *Node*s selected from the *View* by the query statement are called a result set.

Servers may find it difficult to process queries that require access to runtime data, such as device data, that involves resource intensive operations or significant delays. In these cases, the *Server* may find it necessary to reject the query.

7.8 Attribute Service Set

The *Attribute Service Set* is used to read and write *Attribute* values. *Attributes* are primitive characteristics of *Node*s that are defined by OPC UA. They may not be defined by *Clients* or *Servers*. *Attributes* are the only elements in the *AddressSpace* permitted to have data values. A special *Attribute*, the *Value Attribute* is used to define the value of *Variables*.

7.9 Method Service Set

Methods represent the function calls of *Objects*. They are defined in IEC 62541-3. *Methods* are invoked and return after completion, whether successful or unsuccessful. Execution times for *Methods* may vary, depending on the function they are performing.

The *Method Service Set* defines the means to invoke *Methods*. A *Method* is always a component of an *Object*. Discovery is provided through the browse and query *Services*. *Clients* discover the *Methods* supported by a *Server* by browsing for the owning *Objects* that identify their supported *Methods*.

Because *Methods* may control some aspect of plant operations, method invocation may depend on environmental or other conditions. This may be especially true when attempting to re-invoke a *Method* immediately after it has completed execution. Conditions that are required to invoke the *Method* may not yet have returned to the state that permits the *Method* to start again. In addition, some *Methods* may be capable of supporting concurrent invocations, while others may have a single invocation executing at a given time.

7.10 MonitoredItem Service Set

The *MonitoredItem Service Set* is used by the *Client* to create and maintain *MonitoredItems*. *MonitoredItems* monitor *Variables*, *Attributes* and *EventNotifiers*. They generate *Notifications* when they detect certain conditions. They monitor *Variables* for a change in value or status; *Attributes* for a change in value; and *EventNotifiers* for newly generated *Alarm* and *Event* reports.

Each *MonitoredItem* identifies the item to monitor and the *Subscription* to use to periodically publish *Notifications* to the *Client* (see 7.11). Each *MonitoredItem* also specifies the rate at which the item is to be monitored (sampled) and, for *Variables* and *EventNotifiers*, the filter criteria used to determine when a *Notification* is to be generated. Filter criteria for *Attributes* are specified by their *Attribute* definitions in IEC 62541-4.

The sample rate defined for a *MonitoredItem* may be faster than the publishing rate of the *Subscription*. For this reason, the *MonitoredItem* may be configured to either queue all *Notifications* or to queue only the latest *Notification* for transfer by the *Subscription*. In this latter case, the queue size is one.

MonitoredItem Services also define a monitoring mode. The monitoring mode is configured to disable sampling and reporting, to enable sampling only, or to enable both sampling and reporting. When sampling is enabled, the *Server* samples the item. In addition, each sample is evaluated to determine if a *Notification* should be generated. If so, the *Notification* is queued. If reporting is enabled, the queue is made available to the *Subscription* for transfer.

Finally, *MonitoredItems* can be configured to trigger the reporting of other *MonitoredItems*. In this case, the monitoring mode of the items to report is typically set to sampling only, and when the triggering item generates a *Notification*, any queued *Notifications* of the items to report are made available to the *Subscription* for transfer.

7.11 Subscription Service Set

The *Subscription Service Set* is used by the *Client* to create and maintain *Subscriptions*. *Subscriptions* are entities that periodically publish *NotificationMessages* for the *MonitoredItem* assigned to them (see 7.9). The *NotificationMessage* contains a common header followed by a series of *Notifications*. The format of *Notifications* is specific to the type of item being monitored (i.e. *Variables*, *Attributes*, and *EventNotifiers*).

Once created, the existence of a *Subscription* is independent of the *Client’s Session* with the *Server*. This allows one *Client* to create a *Subscription*, and a second, possibly a redundant *Client*, to receive *NotificationMessages* from it.

To protect against non-use by *Clients*, *Subscriptions* have a configured lifetime that *Clients* periodically renew. If any *Client* fails to renew the lifetime, the lifetime expires and the S*ubscription* is closed by the *Server*. When a *Subscription* is closed, all *MonitoredItems* assigned to the *Subscription* are deleted.

Subscriptions include features that support detection and recovery of lost *Messages*. Each *NotificationMessage* contains a sequence number that allows *Clients* to detect missed *Messages*. When there are no *Notifications* to send within the keep-alive time interval, the *Server* sends a keep-alive *Message* that contains the sequence number of the next *NotificationMessage* sent. If a *Client* fails to receive a *Message* after the keep-alive interval has expired, or if it determines that it has missed a *Message*, it can request the *Server* to resend one or more *Messages*.

Bibliography

IEC 62541-12, *OPC Unified Architecture – Part 12: Discovery*[2]

[2] Under consideration.

INTERNATIONAL
ELECTROTECHNICAL
COMMISSION

3, rue de Varembé
PO Box 131
CH-1211 Geneva 20
Switzerland

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