IDTA 02009 OPC UA Server Datasheet

Version 1.0
May 2026

SPECIFICATION

Submodel Template of the
Asset Administration Shell

Submodel Template
IDTA approved
- 100% AAS compliant
- Consistent & interoperable
- Released by the AAS experts

Imprint

Publisher
Industrial Digital Twin Association
Lyoner Str. 18
60528 Frankfurt am Main
Germany
https://www.industrialdigitaltwin.org/

Version history

Date	Version	Comment
04.05.2026	1.0	Release of the official Submodel template published by IDTA.

Contents

Figures

Tables

1 General

1.1 About this Document

This document is a part of a specification series. Each part specifies the contents of a Submodel template for the Asset Administration Shell (AAS). The AAS is described in [1], [2], [3] and [6]. First exemplary Submodel contents were described in [4], while the actual format of this document was derived by the "Administration Shell in Practice" [5]. The format aims to be very concise, giving only minimal necessary information for applying a Submodel template, while leaving deeper descriptions and specification of concepts, structures and mapping to the respective documents [1] to [6].

The target group of the specification are developers and editors of technical documentation and manufacturer information, which are describing assets in smart manufacturing by means of the Asset Administration Shell (AAS) and therefore need to create a Submodel instance with a hierarchy of SubmodelElements. This document especially details on the question, which SubmodelElements with which semantic identification shall be used for this purpose.

1.2 Scope of the Submodel

This Submodel specifies an information model that facilitates the configuration of OPC UA client and provides the identification of an OPC UA Server responsible for the Asset described by the Asset Administration Shell. Based on this information, it is possible to keep the information of an OPC UA server consistent across a device life-cycle.

For the configuration part, OPC UA server datasheet in version 1.0 supports the embedding of *NodeSet* files and other configuration information that are necessary for a client application. It uses the *BuildInfo* object information of the IEC 62541 part 5 [10] specification to identify the server in a cluster of multiple servers. Lastly, OPC UA server datasheet provides endpoint information of already deployed server by using the *EndpointDescription* object information of the IEC 62541 part 4 [9] specification.

This document was developed by a Joint Working Group (JWG) consisting of members of IDTA and OPC UA Foundation.

The usage of the OPC UA server datasheet in operational phase of a device life-cycle is in two parts. First is by native OPC UA client (AAS user application→ green square) in Figure 1 that uses the OPC UA server datasheet Submodel through aasx file or over AAS REST API [7] to understand what the OPC UA server responsible for an asset offers. The other part is by an OPC UA client (asset integration → black square) in Figure 2 that is enabled by Submodel template Asset Interfaces Description (AID).

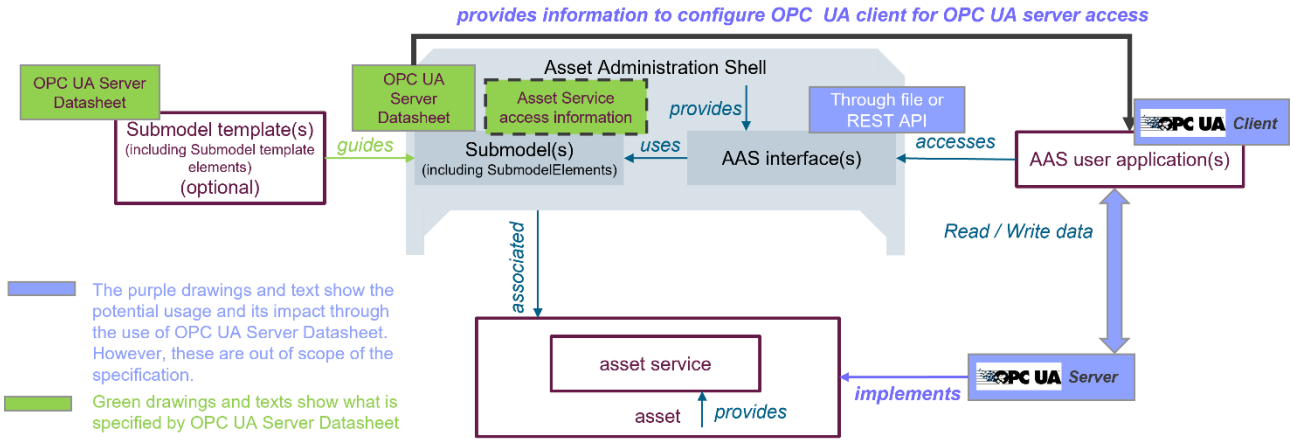

Figure 1: Overall blueprint of OPC UA asset integration with native OPC UA client.

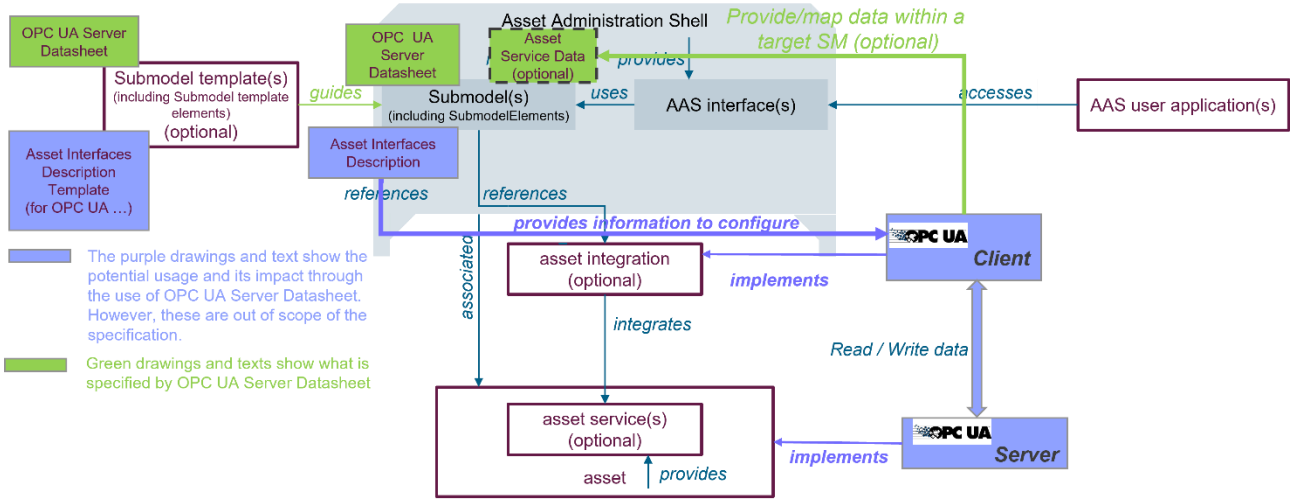

Figure 2: Overall blueprint of OPC UA asset integration with AID.

1.3 Lifecycle usage of OPC UA server datasheet

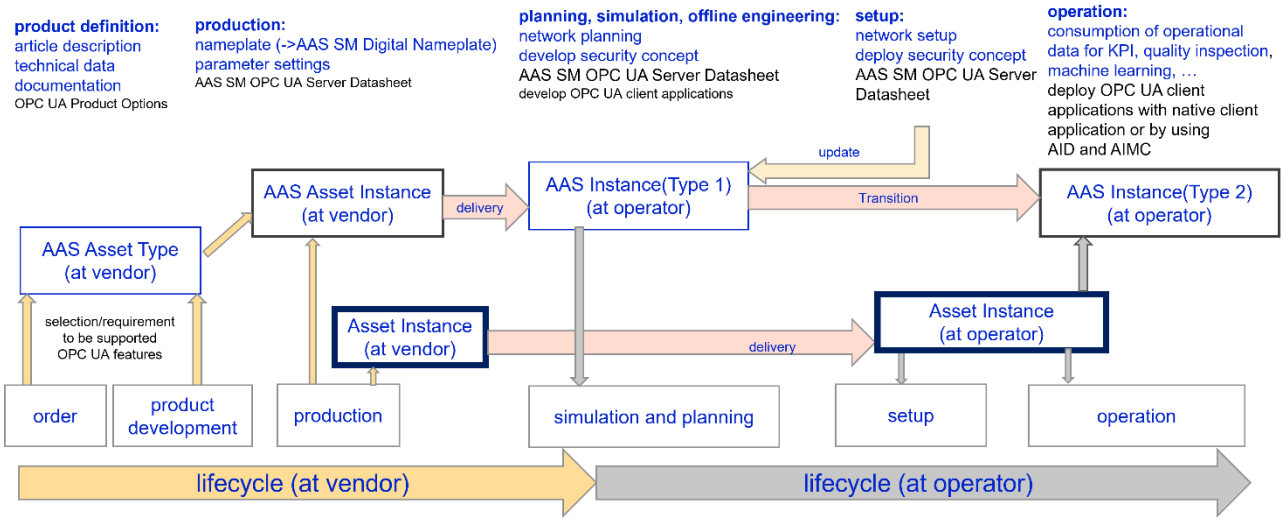

Figure 3: Usage of OPC UA server datasheet across asset lifecycle.

At the inception, when an asset is ordered from a vendor (asset manufacturer), two things are expected to be delivered. The asset and its digital twin which is AAS in this case. Since the delivery of the digital twin can be done faster than the physical asset, this gives the operator a chance to plan and simulate the asset by using its digital twin. Because the asset is designed to use OPC UA server to expose its datapoints to higher level application, then OPC UA server datasheet is expected to be part of the deliverable Submodels for this asset.

At the operator side, when the AAS is delivered with OPC UA server datasheet, it contains configuration and identification information. Part of the configuration information is profiles and facets supported by the server which are nodeset files used to configure the server, supported security modes and policies. Also, for identification, the server can be identified against the BuildInfo Object information (i=2260) of the UA namespace.

When the asset is delivered and properly installed, the next thing is to put it into operation. This brings the asset's lifecycle to setup phase. After deploying the asset over OPC UA server, the OPC UA server datasheet is updated to include server instance information which is endpoint information. This instance information is then used by client applications at the operation phase to access the server and its necessary datapoints.

1.4 Not in Scope of the Submodel

The content of the OPCUA server datasheet submodel can be read and updated by asset's OPC UA server application (asset service in Figure 1 and setup column in Figure 3). The implementation on how this is done is beyond the scope of this document. Also, the implementation of the OPC UA client applications (asset integration or user application in Figure 1) that wants to access the OPC server is beyond the scope of this document.

Also, for the *asset integration* part of Figure 2, the OPC UA server datasheet is a needed extension for asset integration based on OPC UA protocol in addition to submodel templates like Asset Interfaces Description and Asset interfaces Mapping Configuration. Though these submodels are presently published as a protocol agnostic standard template for asset integration, their development and implementation are beyond the scope of this document.

1.5 Relevant Standards for the Submodel Template

- Open Platform Communications Unified Architecture (OPC UA) [8][9][10].

1.6 Use Cases, Requirements and Design Decisions

1.6.1 Use Cases

Table 1: OPC UA Server Datasheet Use Cases

Use Case	Explanation
Device Simulation	1. The machine user ordered an OPC UA enabled machine from machine builder and wants to simulate the operation of the machine with regards to its interface to IT applications. 2. To do this, the machine user needs the digital twin of its machine and the information that the twin will expose to the IT application through its OPC UA interface. 3. The Machine builder provides the digital twin (AAS) of the machine with OPC UA server datasheet Submodel. In this Submodel, information about the nodes that the machine will expose, and the security mechanisms the machine can support are defined. 4. This information allows the machine user to simulate its machine before it is delivered and set up appropriate networks for fast onboarding.
OPC UA client Configuration	1. The operator wants to on-board its machine that has OPC UA server and needs information about what the server offers in a machine-readable way. 2. Part of the information needed is the nodeset file(s) deployed in the server and the list of supported security features. 3. Due to the wide range of OPC UA functionality, features that the asset's OPC UA server support needs to be well defined by the vendor so that the operator can use these features to configure its client application. 4. The machine builder provides this information as OPC UA server datasheet Submodel to the operator. 5. With this information, onboarding the machine at the operator is made easy and fast.
Server Identification	1. In a network with multiple OPC UA servers, identification can play a major role in checking and validating if a connection to the correct server is being established. 2. In OPC UA server datasheet Submodel, the identification information of the server corresponding to a device is provided.

	3. This allows client applications to retrospectively validate the server identification information to what the vendor provided.
Server Access	1. When a device that has integrated OPC UA server is online, it is recommended that the server information is registered at the local discovery server or at the global discovery server. However, this process is not mandatory. If the server application does not use any discovery server, without the operator, it is almost impossible for a client application to know on which endpoint the device is running on and what are its security capabilities. 2. In OPC UA server datasheet the server endpoint description or the discovery endpoint information can be provided. With this information, client applications can understand what the server provides and which way to access the server.

1.6.2 Requirements

- Provide the OPC UA client an understanding of how to access an OPC UA server. Provide the client an understanding what can be expected from an Asset regarding its interface and/or related interface such as which data and functions are served.
- Provide the semantic knowledge and context profiles/facets implemented in the server of such data and functions.
- Provide the client information that is required to retrieve specific data or to use specific functions in terms of protocol settings and security requirements.

1.6.3 Design Decisions

- Define a representation on how and OPC UA server can be identified and accessed as a Submodel.
- Follows some information models define in IEC 62541 part 5 and 9 specification.

2 Submodel OPC UA Server Datasheet

2.1 Approach

The Submodel consists of OPC UA server datasheet information (Figure 4) that specifies information needed to configure an OPC UA client application for Identify and accessing an OPC UA server.

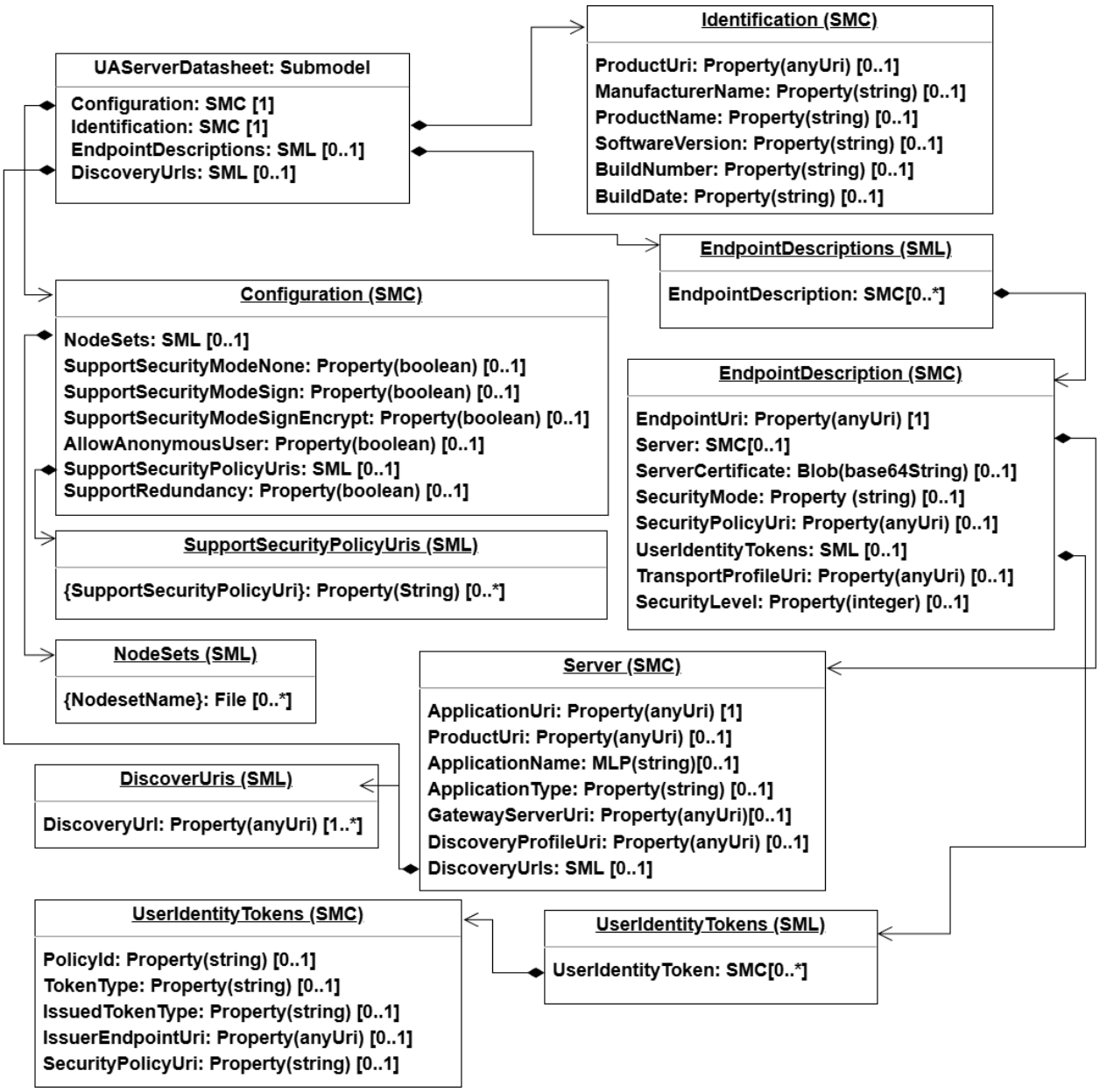

Figure 4: OPC UA server datasheet core structure

2.2 Overview of the AID Core Structure

The properties and collections are grouped into the focus of the Submodel. Each collection at the top-level defines its aim. Configuration SMC is used by the client to understand the capability of the server application, Identification SMC is used for identifying the server. EndpointDescription SMC and DiscoveryUri are used to define how an online server application is configured.

2.3 Elements of the SM “UAServerDatasheet”

Table 2: Attributes of UAServerDatasheetdc Submodel

idShort:	UAServerDatasheet		
Class:	Submodel (SM)		
semanticId:	[IRI] https://admin-shell.io/idta/SubmodelTemplate/opcuaserverdatasheet/1/0		
Parent:	Asset Administration Shell, to which the SM shall be associated to		
Explanation:	Definition of the Submodel OPC UA Server Datasheet identified by its semanticId.		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[SMC] Configuration	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/Configuration Indicates entry point for the configuration parameter of an OPC UA server application.	See Section 2.4	1
[SMC] Identification	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/Identification Indicates the entry point for Identification parameters of an OPC UA server application. The Identification uses BuildInfo structure defined in OPC UA Part 5 [10].	See Section 2.7	1
[SML] EndpointDescriptions	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/EndpointDescriptions Indicates the entry point of an online OPC UA server access. The Endpoint description uses the EndpointDescription structure defined in OPC UA Part 4 [9]. Note: It is recommended that if a server registers at a discovery server, this term can be omitted and the ***DiscoveryUri*** term should be used. But if the server does not register at the discovery server, then it is recommended to use this term to define the endpoint of the online server.	See Section 2.8	0..1
[Property] DiscoveryUrl	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/DiscoveryUrls Provides a list of discovery endpoints used by the server application.	See Section 2.11	0..1

2.4 Elements of the SMC “Configuration”

Table 3: Elements of SMC Configuration

idShort:	Configuration		
Class:	SubmodelElementCollection (SMC)		
semanticId:	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/Configuration		
Parent:	Submodel with idShort = UAServerDatasheet and respective semanticId.		
Explanation:	This SubmodelElementCollection holds the information for an OPC UA server configuration parameter.		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[SML] NodeSets	[IRI]https://admin-shell.io/idta/UAServerDatasheet/1/0/NodeSets Provides a container for nodeset files that is intended for configuring the server. ~~By default, the UA nodeset is always included during configuration as base nodeset.~~	See Section 2.5	0..1
[Property] SupportSecurityModeNone	[IRI]https://admin-shell.io/idta/UAServerDatasheet/1/0/SecurityModeNone Indicates to the client application that a server supports none security mode.	[boolean] true or false	0..1
[Property] SupportSecurityModeSign	[IRI]https://admin-shell.io/idta/UAServerDatasheet/1/0/SecurityModeSign Indicates to the client application that a server supports sign security mode.	[boolean] true or false	0..1
[Property] SupportSecurityModeSignEncrypt	[IRI]https://admin-shell.io/idta/UAServerDatasheet/1/0/SecurityModeSignEncrypt Indicates to the client application that a server supports sign and encrypt security mode.	[boolean] true or false	0..1
[Property] AllowAnonymousUser	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/AllowAnonymous Indicates to the client application that the server allows or denies anonymous users.	[boolean] true or false	0..1
[Property] SupportRedundancy	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/SecurityRedundancy Indicates to the client application that the server supports Redundancy. If the value is true, the type of redundancy supported can be found in *RedundancySupport* node (i=851).	[boolean] True or false	0..1

[SML] SupportSecurityPolicyUris	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/SecurityPolicyUris Provides a container for list of security policies that the server supports.	See	0..1

2.5 Elements of SMC “NodeSets”

Table 4: Elements of SML NodeSets

idShort:	NodeSets		
Class:	SubmodelElementList (SML)		
semanticId:	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/NodeSets		
Parent:	Submodel element collection with idShort = Configuration and respective semanticId.		
Explanation:	This SubmodelElementList holds information about nodeset files that is deployed with the server		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[File] {NodesetName}	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/NodeSetFile Provides a placeholder for a nodeset file that an OPC UA server uses to create *AddressSpace* during configuration. The nodeset can be part of the machine’s AASX file, in an OPC cloud library or in another downloadable endpoint.	[string] /aasx/files/Opc.Ua.Di.NodeSet2.xml Or https://uacloudlibrary.opcuafoundation.org/infomodel/download/1060041392	0..*

2.6 Elements of SML “SupportSecurityPolicyUris”

Table 5 Elements of SML SupportSecurityPolicyUris

idShort:	SupportSecurityPolicyUris		
Class:	SubmodelElementList (SML)		
semanticId:	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/SecurityPolicyUris		
Parent:	Submodel element collection with idShort = Configuration and respective semanticId.		
Explanation:	This SubmodelElementList holds information about nodeset files that is deployed with the server		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[Property] {SecurityPolicyUri}	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/SecurityPolicyUri Indicates the security policy uri supported by the server. For example, if the server supports basic128Rsa15, the value will be http://opcfoundation.org/UA/SecurityPolicy#Basic128Rsa15.	[string] http://opcfoundation.org/UA/SecurityPolicy#Basic128Rsa15.	0..*

2.7 Elements of SMC “Identification”

Table 6: Elements of SMC Identification

idShort:	Identification		
Class:	SubmodelElementCollection (SMC)		
semanticId:	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/Identification		
Parent:	Submodel with idShort = UAServerDataSheet and respective semanticId.		
Explanation:	This SubmodelElementCollection indicates the entry point for Identifying an OPC UA server in a network. It uses the BuildInfo structure defined in OPC UA Part 5 [10].		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[Property] ProductUri	[IRI] 0173-1#02-AAY811#001 This is a URI that identifies the OPC UA server.	[String] urn:KUKA_Deutschland_GmbH:Kuka:OpcUaService	0..1
[Property] ManufacturerName	[IRI] 0173-1#02-AAO677#002 Indicates the name of the OPC UA Server application manufacturer.	[String] KUKA Deutschland GmbH	0..1
[Property] ProductName	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/ProductName Indicates the name of the OPC UA Server application.	[String] Kuka.DeviceConnector	0..1
[Property] SoftwareVersion	[IRI] 0112/2///61987#ABA601#008 Indicates the software version of the OPC UA server application.	[String] 2.15	0..1
[Property] BuildNumber	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/BuildNumber Indicates the build number of the OPC UA server application.	[String] 50	0..1
[Property] BuildDate	[IRI] https://admin-shell.io/idta/SoftwareNameplate/1/0/SoftwareNameplate/SoftwareNameplateType/BuildDate Indicates the build date of the OPC UA Server application as UTC time with format YYYY-MM-DDTHH:MM:SS.sss.	[DateTime] 2025-04-08T09:36:46.649Z	0..1

2.8 Elements of SML “EndpointDescriptions”

Table 7: Elements of SML EndpointDescriptions

idShort:	EndpointDescriptions		
Class:	SubmodelElementList (SML)		
semanticId:	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/EndpointDescriptions		
Parent:	Submodel idShort = UAServerDataSheet and respective semanticId.		
Explanation:	Defines the entry points for an OPC UA server access. The Endpoint description uses the EndpointDescription structure defined in OPC UA Part 4 [9].		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[SMC] {EndpointDescription}	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/EndpointDescription Provides information on how to access an OPC UA server.	See Section 2.9	0..*

2.9 Elements of SMC EndpointDescription

Table 8: Elements of SMC EndpointDescription from SML EndpointDescription

idShort:	EndpointDescription		
Class:	SubmodelElementCollection (SMC)		
semanticId:	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/EndpointDescription		
Parent:	EndpointDescriptions SML		
Explanation:	This SMC holds the information on how to access an OPC UA server.		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[Property] EndpointUri	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/EndpointUri Defines the URL for the server endpoint.	[String] opc.tcp://localhost:5689	1
[SMC] Server	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/Server Defines the description of the server that the endpoint belongs to.	See Section 2.10	0..1
[Blob] ServerCertificate	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/Servercertificate When a server is instantiated, a certificate is issued and the certificate contains information that a client can use to validate a server when establishing a session. The server certificate uses *Application Instance Certificate* model of the OPC UA part 2 [8] and 4[9] as byte64 encoded string.	[Blob]	0..1

[Property] SecurityMode	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/SecurityMode Defines the type of security mode attached to the endpoint description. The mode can be one of the following values: None, Sign, SignAndEncrypt, Invalid.	[String] Sign	0..1
[Property] SecurityPolicyUri	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/SecurityPolicyUri Defines the security policy to use when securing messages.	[String] http://opcfoundation.org/UA/SecurityPolicy#Basic256Sha256	0..1
[SML] UserIdentityTokens	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/UserIdentityTokens Defines the list of user identity tokens that the server will accept.	See Section 2.12	0..1
[Property] TransportProfileUri	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/TransportProfileUri Defines the URI of the transport profile supported by the server endpoint. Because the transport profiles can be updated in the future, this value is intended to support both future and present transport profile uri officially defined by OPC UA.	[String] http://opcfoundation.org/UA-Profile/Transport/uatcp-uasc-uabinary	0..1
[Property] SecurityLevel	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/SecurityLevel A numeric value that indicates how secure the EndpointDescription is compared to other EndpointDescriptions for the same Server. A value of 0 indicates that the EndpointDescription is not recommended and is only supported for backward compatibility. A higher value indicates better security.	[Integer] • 0 = Weak Security • >0 = Better Security	0..1

2.10 Elements of SMC “Server“

Table 9: Element of SMC Server

idShort:	Server		
Class:	SubmodelElementCollection (SMC)		
semanticId:	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/Server		
Parent:	Submodel element collection with idShort = EndpointDescription and respective semanticId.		
Explanation:	Defines the description of the server that the endpoint belongs to.		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[Property] ApplicationUri	[IRI] 0173-1#02-AAY811#001 Defines a URI that uniquely identifies the application instance.	[String] urn:opcuademonstrator:kuka:OpcUaService	1

[Property] ProductUri	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/ProductUri This is a URI that identifies the OPC UA server.	[String] urn:KUKA_Deutschland_GmbH:Kuka:OpcUaService	0..1
[MultiLanguageProperty] ApplicationName	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/ApplicationName Defines a localized descriptive name for the server application.	[String] en: KUKA.Deviceconnector.OpcUaService	0..1
[Property] ApplicationType	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/ApplicationType Defines the type of application as an enumeration. Acceptable values for this term are Server, Client, ClientAndServer, DiscoveryServer. For UAServerDataSheet Submodel, the focus is on OPC UA server so only Server and ClientAndServer value are valid for this Submodel.	[String] Server	0..1
[Property] GatewayServerUri	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/GatewayServerUri Defines the URI that identifies the gateway server that is linked with the discovery Uris. If the server can be accessed directly, this term is not defined.	[String]	0..1
[Property] DiscoveryProfileUri	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/DiscoveryProfileUri Defines the URI that identifies the discovery profile that is supported by the URLs.	[String]	0..1
[SML] DiscoveryUrls	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/DiscoveryUrls Provides a list of discovery endpoints used by the server application.	See Section 2.11	0..1

2.11 Elements of SML “DiscoveryUrls”

Table 10: Elements of SML DiscoveryUrls

idShort:	DiscoveryUrls		
Class:	SubmodelElementList (SML)		
semanticId:	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/DiscoveryUrls		
Parent:	Server SMC		
Explanation:	This SML holds the list of discovery Urls used by the server application.		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[Property] DiscoveryUrl	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/DiscoveryUrl Defines the discovery Url of the server application.	[String] opc:tcp://OpcUaDemonstrator:4840/	0..*

2.12 Elements of SML “UserIdentityTokens“

Table 11: Element of SML UserIdentityTokens

idShort:	UserIdentityTokens		
Class:	SubmodelElementList (SML)		
semanticId:	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/UserIdentityTokens		
Parent:	EndpointDescription SMC		
Explanation:	Contains the list of user identity tokens that the server will accept.		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[SMC] UserIdentityToken	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/UserIdentityToken Defines an identity token that can be used to access the server application.	See section 2.13	0..*

2.13 Elements of SMC UserIdentityToken

Table 12: Elements of SMC UserIdentityToken

idShort:	UserIdentityToken
Class:	SubmodelElementCollection (SMC)
semanticId:	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/UserIdentityToken
Parent:	UserIdentityTokens SML
Explanation:	This SMC holds the information of an identity token that the server can accept during active session request.

[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[Property] PolicyId	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/PolicyId PolicyId is provided by a server application to defines an identifier for the token policy.	[string] Certificate.Basic256Sha256-Sign	0..1
[Property] TokenType	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/TokenType Indicates the kind of user identity token required. This term us an enumeration with one of the following values Anonymous,Username,Certificate,IssueToken.	[string] • Anonymous • Username • Certificate • IssuedToken	0..1
[Property] IssuedTokenType	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/IssuedTokenType Specified only when *TokenType* is *IssuedToken* to indicates the URI for the type of token.	[string]	0..1
[Property] IssuerEndpointUrl	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/IssuerEndpointUri Defines the endpoint of the authorization service. The value provided here depends on the *IssuedTokenType* term.	[string]	0..1
[Property] SecurityPolicyUri	[IRI] https://admin-shell.io/idta/UAServerDatasheet/1/0/SecurityPolicyUri Defines the security policy to use when encrypting or signing the UserIdentityToken when it is passed to the Server in the active session request.	[string]	0..1

Annex A. Document Table Formats

1. General

The tables used in this document try to outline information as concisely as possible. They do not convey all information on Submodels and SubmodelElements. For this reason, the definitive definitions are given by a separate file in form of an AASX file of the Submodel template and its elements.

2. Tables of Submodels and SubmodelElements

For clarity and brevity, a set of rules is used for the tables for describing Submodels and SubmodelElements.

- The tables follow in principle the same conventions as in [5].
- The table heads abbreviate 'cardinality' with 'card'.
- The tables often place two information in different rows of the same table cell. In this case, the first information is marked out by sharp brackets [] form the second information. A special case are the semanticIds, which are marked out by the format: (type)(local)[idType]value.
- The types of SubmodelElements are abbreviated:

SME type	SubmodelElement type
Property	Property
MLP	MultiLanguageProperty
Range	Range
File	File
Blob	Blob
Ref	ReferenceElement
Rel	RelationshipElement
SMC	SubmodelElementCollection
SML	SubmodelElementList

- If an idShort ends with '__00__', this indicates a suffix of the respective length (here: 2) of decimal digits, in order to make the idShort unique. A different idShort might be choosen, as long as it is unique in the parent's context.
- The Keys of semanticId in the main section feature only idType and value, such as: [IRI]https://admin-shell.io/vdi/2770/1/0/DocumentId/Id. The attributes "type" and "local" (typically "ConceptDescription" and "(local)" or "GlobalReference" and (no-local)") need to be set accordingly; see [6].
- If a table does not contain a column with "parent" heading, all represented attributes share the same parent. This parent is denoted in the head of the table.
- Multi-language strings are represented by the text value, followed by '@'-character and the ISO 639 -1 language code: example@en.
- The [valueType] is only given for Properties.

Annex B. UAServerDatasheet in AASX Package Explorer

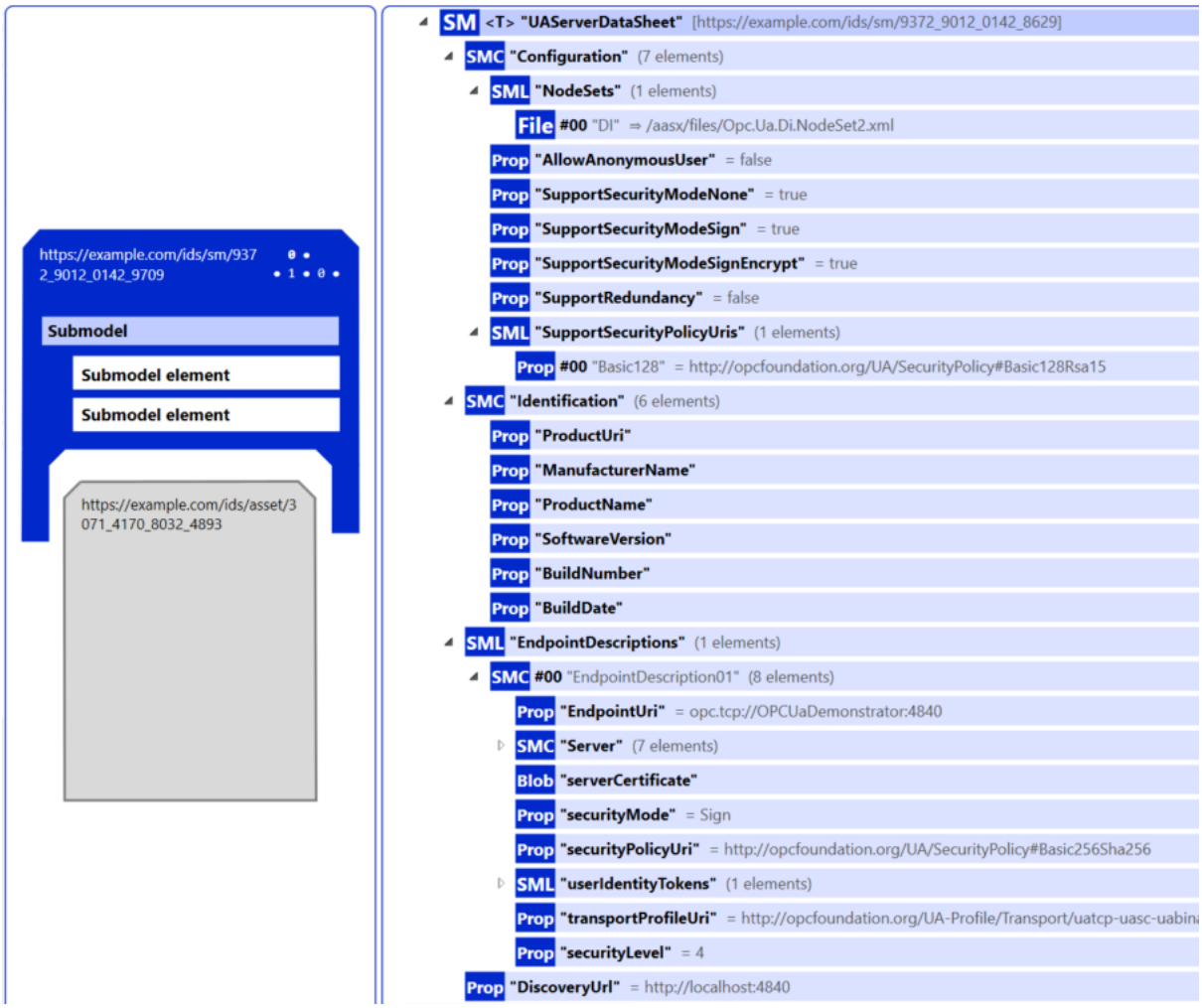

Figure 5: Example description of a device AAS with OPC UA Server Datasheet Submodel.

Bibliography

[1] "Recommendations for implementing the strategic initiative INDUSTRIE 4.0", acatech, April 2013. [Online]. Available https://www.acatech.de/Publikation/recommendations-for-implementing-the-strategic-initiative-industrie-4-0-final-report-of-the-industrie-4-0-working-group/

[2] "Implementation Strategy Industrie 4.0: Report on the results of the Industrie 4.0 Platform"; BITKOM e.V. / VDMA e.V., /ZVEI e.V., April 2015. [Online]. Available: https://www.bitkom.org/noindex/Publikationen/2016/Sonstiges/Implementation-Strategy-Industrie-40/2016-01-Implementation-Strategy-Industrie40.pdf

[3] "The Structure of the Administration Shell: TRILATERAL PERSPECTIVES from France, Italy and Germany", March 2018, [Online]. Available: https://www.plattform-i40.de/I40/Redaktion/EN/Downloads/Publikation/hm-2018-trilaterale-coop.html

[4] "Beispiele zur Verwaltungsschale der Industrie 4.0-Komponente – Basisteil (German)"; ZVEI e.V., Whitepaper, November 2016. [Online]. Available: https://www.zvei.org/presse-medien/publikationen/beispiele-zur-verwaltungsschale-der-industrie-40-komponente-basisteil/

[5] "Verwaltungsschale in der Praxis. Wie definiere ich Teilmodelle, beispielhafte Teilmodelle und Interaktion zwischen Verwaltungsschalen (in German)", Version 1.0, April 2019, Plattform Industrie 4.0 in Kooperation mit VDE GMA Fachausschuss 7.20, Federal Ministry for Economic Affairs and Energy (BMWi), Available: https://www.plattform-i40.de/PI40/Redaktion/DE/Downloads/Publikation/2019-verwaltungsschale-in-der-praxis.html

[6] "Specification of the Asset Administration Shell; Part 1 - Metamodel", March 2023, Industrial Digital Twin Association, Available: https://industrialdigitaltwin.org/wp-content/uploads/2023/06/IDTA-01001-3-0_SpecificationAssetAdministrationShell_Part1_Metamodel.pdf

[7] "Specification of the Asset Administration Shell; Part 2 – Application Programming Interfaces", April 2023, Industrial Digital Twin Association, Available: https://industrialdigitaltwin.org/wp-content/uploads/2023/04/IDTA-01002-3-0_SpecificationAssetAdministrationShell_Part2_API.pdf

[8] IEC 62541-2: OPC Unified Architecture Security Model, Edition 3.0; OPC Foundation; International Electrotechnical Commission. November, 2020.

[9] IEC 62541-4: OPC Unified Architecture Services, Edition 3.0; OPC Foundation; International Electrotechnical Commission. November, 2020.

[10] IEC 62541-5: OPC Unified Architecture Information Model, Edition 3.0; OPC Foundation; International Electrotechnical Commission. July, 2020.

www.industrialdigitaltwin.org