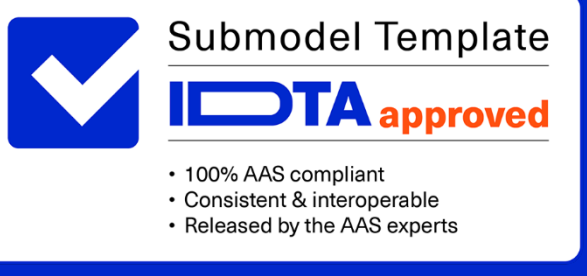
IDTA 02017-1-0 Asset Interfaces Description

January 2024

SPECIFICATION

Submodel Template of the Asset Administration Shell



Imprint

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

Version history

Date	Version	Comment
19.01.2024	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 and a common representation for describing the interface(s) of an asset service or asset related service. Based on this information, it is possible to initiate a connection to such kind of service and start to request or subscribe to served datapoints, and/or perform operations. Such datapoints of a system service can be, for example, various sensor and/or status values, and an operation can trigger an actuator, such as switching a motor “on” or “off”.

The Asset Interfaces Description (AID) in version 1.0 supports the description of interfaces based on three specific protocols: Modbus, HTTP and MQTT. Any other protocols and interfaces will be addressed in upcoming versions of the AID. As a forward-looking note, the AID working group at IDTA has outlined plans for the AID 1.1 version to incorporate support for both OPC UA (joint activity with the OPC Foundation) and BACnet.

The W3C Web of Things Thing Description (WoT TD) as an open, royalty-free standard is considered as a baseline for the content and structure of the definition of this Submodel template.

In addition to the protocol specific information provided by the AID, it also provides the ability to reference external descriptors such as GSD, GSDML, IO Device Description, WoT TD (as a supplement) etc. These external descriptor is not restricted to the protocols currently defined in the AID.

As a complement to the AID, an Asset Interfaces Mapping Configuration (AIMC) Submodel can be used to map the received data from the asset services to a specific place within an AAS (e.g. an application specific Submodel to monitor data). The principle scope and use of the AID Submodel in combination with an AIMC is explained in the following figure:

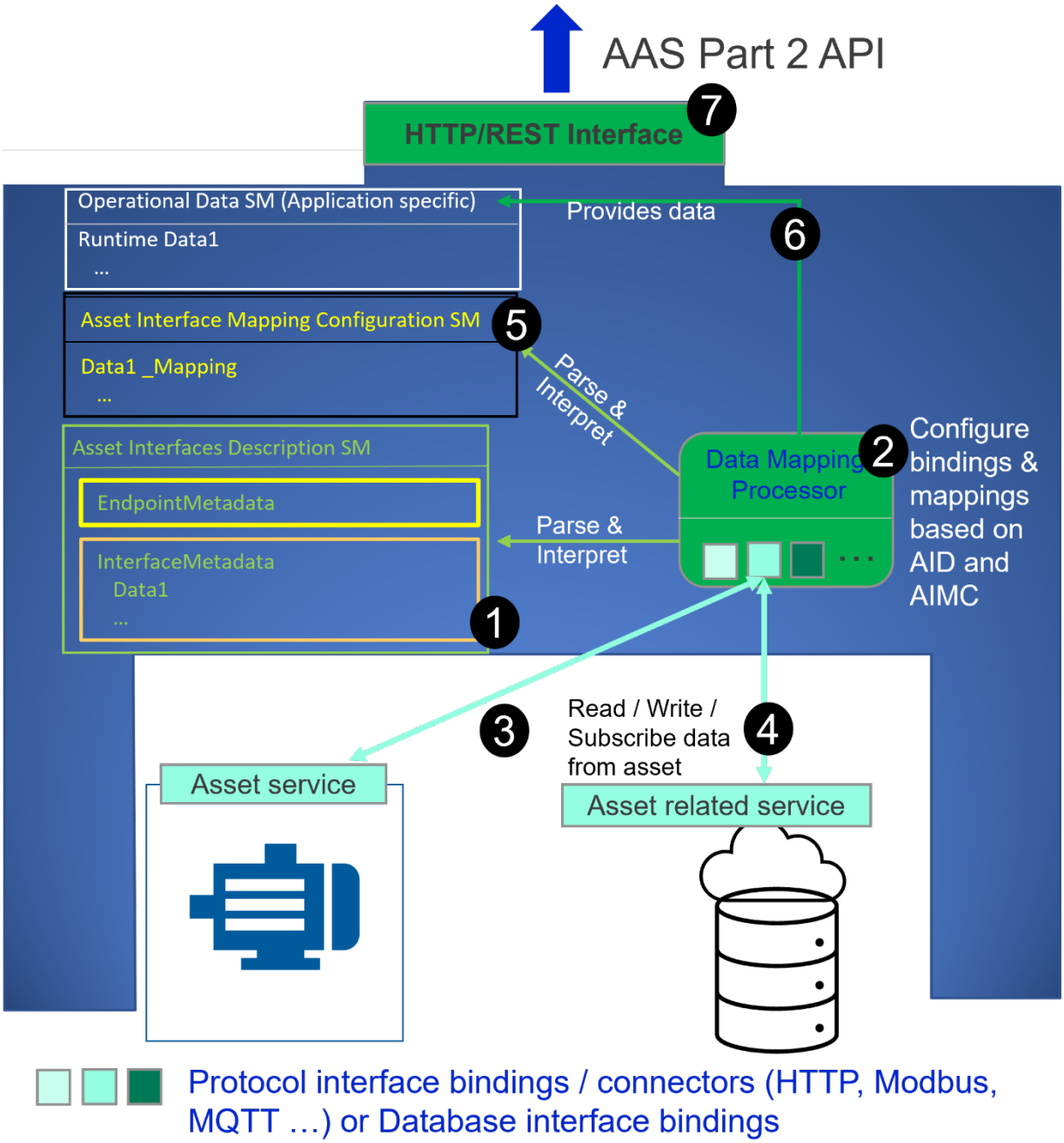


Figure 1 Principle AID submodel usage and possible data mapping process e.g. by Asset Interfaces Mapping Configuration (AIMC).

The number legends in Figure 1 are described as follows:

(1) Asset Interfaces Description Submodel: it holds the description model of the asset service (or asset related service) interfaces and its datapoint.

(2) Data Mapping Processor (DMP): This is a software component that provides connection (e.g., via Modbus) to the asset service and/or asset related service and exchanges data as defined within the AID Submodel. It also manages the mapping of retrieved data to a desired SM according to AIMC SM definition.

(3) Data transmission channel between Data Mapping Processor and asset service. Depending on the underlying protocol (e.g. Modbus, MQTT) used by the asset service (and as described by the AID), the specific datapoint can be requested/subscribed.

(4) Data transmission channel between Data Mapping Processor and asset related service. Depending on the underlying protocol (e.g. HTTP) used by the asset related service (and as described by the AID), the specific datapoint can be requested/subscribed.

(5) AIMC Submodel: it provides the necessary information about the mapping of the datapoints described by the AID to elements in a desired (application-specific) operation data Submodel.

(6) Operational Data Submodel: it is a Submodel where the (runtime) data is being stored. The details about this location are in the AIMC. With AIMC's information, the Data Mapping Processor can correctly map the asset's data to the right parts of the Submodel.

(7) HTTP/REST Interface: This is an AAS Interface defined in details of AAS Part 2 as a standardized API [11]. It is used to enable communication between AASX server and external applications.

1.3 Not in Scope of the Submodel

The asset services or asset related services described in AID are parsed and interpreted by an application (e.g. Data Mapping Processor). This application is beyond the scope of this document, but its main functionality is to handle data transmission between an AAS and its asset service or asset related service. It can be packaged with several communication protocol adapters and database integration drivers.

Out of the scope of the AID 1.0 is the detailed definition of actions and events of asset interfaces. The AID 1.0 focuses on monitoring purposes and thus concentrates on properties definitions. The actions and events paradigm will be introduced in one of the forthcoming AID versions.

1.4 Relevant Standards for the Submodel Template

- W3C Web of Things Thing Description (WoT TD) [7]
- Modbus [8]
- MQTT [9]
- HTTP [10]
- IDTA Asset Interfaces Mapping Configuration (AIMC) Submodel

1.5 Use Cases, Requirements and Design Decisions

1.5.1 Use Cases

Table 1: AID Use Cases

Use Case	Explanation
Device & datapoint onboarding	1. The engineer wants to import for example different sensor measurements from energy meters of different vendors into a backend system to monitor energy data of a plant system 2. Each energy meter may support different protocols (e.g., Modbus, Profibus, HTTP) and data model representation 3. Depending on the industry sector and supported protocol energy meters may have an interface description existing such as EDDL that can be used for onboarding purposes 4. However, if such electronic description doesn't exists, typically the communication & datapoint metadata is retrieved from manuals (PDF, webpages, ...) or tables (CVS, Excel,...) 5. Taking over such information in application or engineering development is an expensive process (e.g., c&p over 80 datapoints of a single device) and causes risk

	of error proneness (e.g., wrong calculation or interpretation of communication metadata)
Asset Data Manipulation	1. The Target User connects to the AAS of the Asset of interest 2. The Target User reads the Asset Interfaces Description Submodel describing which interface of the Asset can be selected (e.g., a Modbus-based or MQTT-based interface) with which security requirements 3. The Target User selects an interface of the Asset (e.g., Modbus-based interface) 4. The Target User configures a client for the relevant Asset interface fulfilling the security requirements 5. The Target User deploys the configured client that connects to the Asset interface and initiate a specific action of the asset (e.g., change a state)
Asset Data Integration	1. The Target User connects to the AAS of the Asset of interest 2. The Target User connects to the Asset Interfaces Description Submodel describing which datapoints the Asset provides through which interfaces with which security requirements 3. The Target User selects relevant datapoints from the relevant Asset interfaces 4. The Target User configures one client per relevant Asset interface fulfilling the security requirements and configuring a mapping (via an Asset Interface Mapping Configuration Submodel) of the selected Asset's data into a data sink such as in a separate Submodel 5. The Target User deploys the configured clients that connect to the Asset interfaces and integrates data into the specified data sinks Example: • An application-specific OperationData Submodel is intended to provide measurement data of a Sensor-Asset • The AID Submodel specify Sensor-Asset's interface (it describes which measurement datapoints can be read/subscribed, which protocol is used and which security requirements must be met) • An Asset Interface Mapping Configuration (AIMC) defines the mapping where the read runtime measurement datapoints of the Sensor-Asset should be represented/mapped within the application-specific OperationData Submodel (e.g., to a SME Property element)

1.5.2 Requirements

- 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 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.
- Provide required information that runtime data can be mapped to a specific place (e.g., to an AAS SME Property) within a SM.
- To reuse existing concepts and standards the AID shall be derived from WoT Thing Description (WoT TD) specification as a common protocol-agnostic approach.
- Different industry domains use different established device descriptors (e.g., GSD, GSDML, EDD, FDI Packages, Instanced OPC UA Companion Specifications, etc.). AID shall provide the opportunity to refer to such existing device descriptors.
- In the case a WoT TD already exists, it is recommended that an AID instance provides a reference to the WoT TD. A WoT TD can be used to enable more knowledge about the interface that is possible by the WoT TD specification (e.g., the Web Linking concepts or read-all-properties feature).

1.5.3 Design Decisions

- Define a common representation of the asset's (related) interfaces as a SM.
- Follow WoT TD with its model structure as an open standard with its parameter terms to describe the endpoint details and interface metadata of an asset. It is recommended that the WoT TD specification be used as supplemental literature (also see [7]) to the AID specification for additional background information and examples.
- Each specified term in W3C WoT TD has a namespace definition. To avoid misinterpretation and duplicate semantic definitions, the most assigned semanticId in the AID is based on the namespace term definition of WoT TD.
- Provide a specific place to enable the inclusion of existing device or interface descriptors.
- If a description file (WoT TD, GSDML, MTP etc.) of an asset interface exist in the Submodel ExternalDescriptor, the specified mandatory clause for EndpointMetadata SMC and InteractionMetadata SMC should be seen as optional.

2 Submodel Asset Interfaces Description

2.1 Approach

The Submodel consists of an AID core part that specifies the basic structure of the AID Submodel, which is identical regardless of the type of protocol interface (e.g., Modbus, HTTP). There will be also a common set of properties and collections definitions.

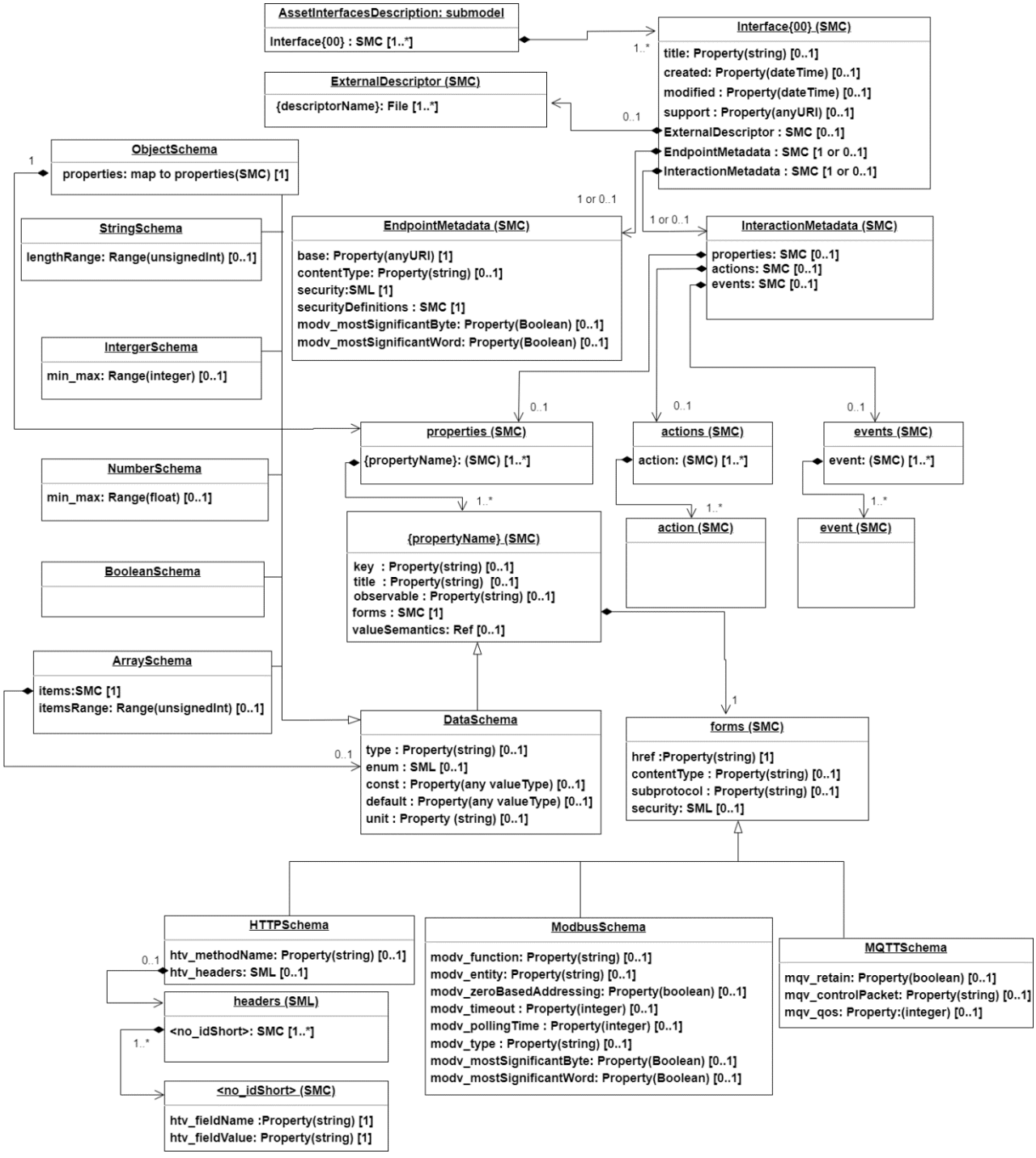


Figure 2 AID core structure

Various properties and collections are specified, depending on the protocol used that underlies an interface. For example, Modbus will specify all information to address, e.g., register information and byte length. HTTP provides the HTTP methods such as GET, POST and additional HTTP header information. As seen from the AID core structure, the W3C Web of Things Thing Description (TD) structure and protocol bindings vocabularies are strongly adopted throughout this document for protocol dependent properties.

2.2 Overview of the AID Core Structure

An AID Submodel describes one or more interfaces in form of a SMC. Each SMC of an interface will provide information about its endpoint and interface metadata. The endpoint metadata provides information about how the interface can be reached, e.g., by specifying the IP address or domain name and endpoint path where the asset's server is running. In addition, it also provides information about the default serialization format used when data is exchanged (e.g., JSON, XML, octet stream). If needed, a place holder (ExternalDescriptor) for additional document and mechanism that would be used ease asset connection is provided.

2.3 Elements of the SM "AssetInterfacesDescription"

Table 2: Attributes of AssetInterfacesDescription submodel

idShort:	AssetInterfacesDescription Note: a different idShort might be used, as long as it is unique in the Submodel.		
Class:	Submodel (SM)		
semanticId:	[IRI] https://admin-shell.io/idta/AssetInterfacesDescription/1/0/Submodel		
Parent:	Asset Administration Shell, to which the SM shall be associated to		
Explanation:	Definition of the Submodel Asset Interfaces Description identified by its semanticId. The Submodel idShort can be picked freely.		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[SMC] Interface{00}	[IRI] https://admin-shell.io/idta/AssetInterfacesDescription/1/0/Interface supplementalSemandicId: [IRI] (only if modbus is used) http://www.w3.org/2011/modbus [IRI] (only if mqtt is used) http://www.w3.org/2011/mqtt [IRI] (only if http is used) http://www.w3.org/2011/http [IRI] https://www.w3.org/2019/wot/td Indicates entry point for a particular asset interface description based on Modbus, MQTT, or HTTP (indicated by its semanticId).	Interface00 Interface_MQTT Modbus	1..*

2.4 Elements of the SMC “Interface”

Table 3: Elements of SMC Interface

idShort:	Interface{00} Note: a different idShort might be used, as long as it is unique in the Submodel.		
Class:	SubmodelElementCollection (SMC)		
semanticId:	[IRI] https://admin-shell.io/idta/AssetInterfacesDescription/1/0/Interface supplementalSemandicId: [IRI] (only if modbus is used) http://www.w3.org/2011/modbus [IRI] (only if mqtt is used) http://www.w3.org/2011/mqtt [IRI] (only if http is used) http://www.w3.org/2011/http [IRI] https://www.w3.org/2019/wot/td		
Parent:	Submodel with idShort = AssetInterfacesDescription and respective semanticId.		
Explanation:	This SubmodelElementCollection holds the information for EndpointMetadata and InteractionMetadata. Note: The Interface SMC may also be used to describe interfaces with protocols not yet covered by the AID yet (e.g., only the ExternalDescriptor will be used for providing a GSDML reference for a Profinet communication). In such a case, an appropriate supplementalSemanticId is recommended to identify the purpose of this interface.		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[Property] title	[IRI]https://www.w3.org/2019/wot/td#title Provides a human-readable title to give a human-readable context of the interface.	[string] Robot Modbus Interface	1
[Property] created	[IRI]http://purl.org/dc/terms/created Provides information when the AID Submodel was created.	[string] 2022-12-27 08:26:49.219717	0..1
[Property] modified	[IRI]http://purl.org/dc/terms/modified Provides information when the AID Submodel was modified.	[string] 2022-12-27 08:26:49.219717	0..1
[Property] support	[IRI]https://www.w3.org/2019/wot/td#supportContact Provides an address on how to contact the maintainer of AID Submodel as URI scheme.	[string] mailto:aidsupport@idta.com	0..1

[SMC] EndpointMetadata	[IRI]https://admin-shell.io/idta/AssetInterfacesDescription/1/0/EndpointMaetadata Provides the metadata of the asset's endpoint (base, content type that is used for interaction, etc)	See Section 2.5	1 or 0..1
[SMC] InteractionMetadata	[IRI]https://admin-shell.io/idta/AssetInterfacesDescription/1/0/InteractionMetadata supplementalSem.Id: [IRI]https://www.w3.org/2019/wot/td#InteractionAffordance Provides the metadata of the actually interfaces such as which datapoints and functions are provided by the properties, actions, and events interaction abstraction.	See Section 2.6	1 or 0..1
[SMC] ExternalDescriptor	[IRI]https://admin-shell.io/idta/AssetInterfacesDescription/1/0/ExternalDescriptor Provides a place for existing description files (e.g., Thing Description, GSDML, etc,).	See Section 2.7	0..1

2.5 Elements of SMC “EndpointMetadata“

Table 4: Elements of SMC EndpointMetadata

idShort:	EndpointMetadata
Class:	SubmodelElementCollection (SMC)
semanticId:	[IRI]https://admin-shell.io/idta/AssetInterfacesDescription/1/0/EndpointMetadata
Parent:	Submodel element collection with idShort = Interface{00} and respective semanticId.
Explanation:	This SubmodelElementCollection holds information about asset’s entry point, security and data format serialization.

[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[Property] base	[IRI] https://www.w3.org/2019/wot/td#baseURI Defines asset connection entry point. Each protocol specifies a base pattern. Please see Annex B.1 for more details.	[string] modbus+tcp://192.168.99.159:502/	1
[Property] contentType	[IRI]https://www.w3.org/2019/wot/hypermedia#forContentType Defines content type based on a media type (e.g., text/plain) and potential character decoding/encoding type (e.g., charset=utf-8) for the media type (see RFC2046) of the whole interface. Note: There is also an optional contentType at the forms level (see Section 2.12) within an, e.g., interaction property. The local contentType definition (in the case it is used) overwrites this global contentType definition.	[string] application/json	0..1
[SMC] securityDefinitions	[IRI]https://www.w3.org/2019/wot/td#definesSecurityScheme Defines the security scheme according to W3C: • BasicSecurityScheme (basic_sc) • DigestSecurityScheme (digest_sc) • APIKeySecurityScheme (apikey_sc) • BearerSecurityScheme (bearer_sc) • PSKSecurityScheme (psk_sc) • OAuth2SecurityScheme (oauth2_sc) • AutoSecurityScheme (auto_sc) • NoSecurityScheme (nosec_sc) Note: Even if the interface does not support any security mechanisms, the securityDefinitions should be used explicitly with a nosec_sc entry. Note2: If one of the default defined security scheme does not fulfill the security requirement of the interface, an application-specific securityDefinitions scheme can be introduced and be used in the AID. Here it should be aware, that a clear semanticId context should be applied for identification of this application-specific scheme.	See Section 2.15 securityDefinitions |_basic_sc |_...	1

[SML] security	[IRI]https://www.w3.org/2019/wot/td#hasSecurityConfiguration Selects one or more of the security scheme(s) that can be applied at runtime from the collection of security schemes defines in securityDefinitions. Note: Even if the interface does not support any security mechanisms, the security should have a list entry with a reference to a nosec_sc security definition.	See Section 2.14 security[Ref to basic_sc in securityDefinitions]	1
[Property] modv_mostSignificantByte	[IRI]https://www.w3.org/2019/wot/modbus#hasMostSignificantByte This property is only applicable for Modbus-based communication. When modv_mostSignificantByte is true, it describes that the byte order of the data in the Modbus message is the most significant byte first (i.e., Big-Endian). When false, it describes the least significant byte first (i.e., Little-Endian).	[boolean] true	0..1
[Property] modv_mostSignificantWord	[IRI]https://www.w3.org/2019/wot/modbus#hasMostSignificantWord This property is only applicable for Modbus-based communication. When modv_mostSignificantWord is true, it describes that the word order of the data in the Modbus message is the most significant word first (i.e., no word swapping). When false, it describes the least significant word first (i.e. word swapping).	[boolean] true	0..1

2.6 Elements of SMC “InteractionMetadata“

Table 5: Elements of SMC InteractionMetadata

idShort:	InteractionMetadata
Class:	SubmodelElementCollection (SMC)
semanticId:	[IRI]https://www.w3.org/2019/wot/td#InteractionAffordance
Parent:	Submodel element collection with idShort = Interface{00} and respective semanticId.
Explanation:	This SubmodelElementCollection holds the information of the interaction affordances with properties, actions, and events. An interaction property exposes typically state as datapoint via asset’s interface. This state can then be retrieved (read) and/or observed (subscription). An interaction action allows to invoke a function via asset’s interface, which manipulates state (e.g., toggling a lamp on or off) or triggers a process on the asset (e.g., dim a lamp over time). An interaction event describes an event source via asset’s interface, which asynchronously pushes event data to receivers (e.g., overheating alerts).

[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[SMC] properties	[IRI]https://www.w3.org/2019/wot/td#PropertyAffordance Collection of asset’s datapoint definitions as property SMC (also see Section 2.8).	properties |_status |_voltage |_...	0..1
[SMC] actions	[IRI]https://www.w3.org/2019/wot/td#ActionAffordance Collection of functions that can be done on asset as action SMC	actions |_onOff |_fadeIn |_...	0..1
[SMC] events	[IRI]https://www.w3.org/2019/wot/td#EventAffordance Collection of events triggerable by datapoint state as event SMC	events |_overheading |_alarm1 |_...	0..1

2.7 Elements of SMC “ExternalDescriptor“

Table 6: Elements of SMC ExternalDescriptor

idShort:	ExternalDescriptor		
Class:	Submodel (SM)		
semanticId:	[IRI] https://admin-shell.io/idta/AssetInterfacesDescription/1/0/ExternalDescriptor		
Parent:	Submodel element collection with idShort = Interface{00} and respective semanticId.		
Explanation:	Provides a place for existing description files (e.g., Thing Description, GSDML, etc,).		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[File] {descriptorName}	[IRI]https://admin-shell.io/idta/AssetInterfacesDescription/1/0/externalDescriptorName File reference (local in AASX or outside) to an external descriptor description (e.g., Thing Description, GSDML, MTP, etc,).	• [string] ./sensor_device.td.jsonld • [string] ./gsdml-v21-ed2.xml • [string] https://example.com/myDescriptorFile	1..*

2.8 Elements of SMC “properties“

Table 7: Element of SMC properties

idShort:	properties		
Class:	SubmodelElementCollection (SMC)		
semanticId:	[IRI] https://www.w3.org/2019/wot/td#hasPropertyAffordance		
Parent:	Submodel element collection with idShort = InteractionMetadata and respective semanticId.		
Explanation:	This SubmodelElementCollection collects the interaction affordance properties.		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[SMC] {property_name}	[IRI] https://admin-shell.io/idta/AssetInterfacesDescription/1/0/PropertyDefinition supplementalSemandicId: [IRI] https://www.w3.org/2019/wot/td#name Defines an interaction property that covers usually a datapoint definition that can be read or subscribed to.	See Section 2.9 • [idShort] pump_speed • [idShort] TemperatureValue	0..*

2.9 Elements of SMC “{property_name}”

Table 8: Elements of SMC {property_name}

idShort:	{property_name} Note: {property_name} is an abstract name (e.g., “rotation_speed”) for an interaction property that includes a specific datapoint of an asset. {property_name} will detail the specifics of the datapoint (e.g., data type, restrictions, and semantics) and also explain the underlying communication protocol (e.g., Modbus) that governs how this interaction property and its datapoint can be read or subscribed to.
Class:	SubmodelElementCollection (SMC)
semanticId:	[IRI] https://admin-shell.io/idta/AssetInterfacesDescription/1/0/PropertyDefinition supplementalSemandicId: [IRI] https://www.w3.org/2019/wot/td#name
Parent:	Submodel element collection with idShort = properties and respective semanticId.
Explanation:	This SubmodelElementCollection defines characteristics of an interaction affordances with its datapoint specifications and how to address it via a specific protocol (e.g., Modbus register).

[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[Property] key	[IRI] https://admin-shell.io/idta/AssetInterfacesDescription/1/0/key Optional element when the idShort of {property_name} cannot be used to reflect the desired property name due to the idShort restrictions (e.g., payload message uses “temperature-value” as key term).	[string] temperature-value	0..1
[Property] title	[IRI]https://www.w3.org/2019/wot/td#title Provides a human-readable title of this interaction (e.g., display a text for UI representation)	[string] Rotation speed	0..1
[Property] observable	[IRI] https://www.w3.org/2019/wot/td#isObservable An indicator that tells that the interaction datapoint can be observed with a, e.g., subscription mechanism by an underlying protocol. In case of MQTT, it is recommended that observable=true for each interaction property.	[boolean] true	0..1
[SMC] forms	[IRI] https://www.w3.org/2019/wot/td#hasForm Contains information about datapoint resource location. Note, forms is only available at the top level {property_name}	See Section 2.12	1
[Property] type	[IRI] https://www.w3.org/1999/02/22-rdf-syntax-ns#type Indicates the abstract data type (one of object, array, string, number, integer, boolean, or null) of the described datapoint.	[string] integer	0..1

[Property] const	[IRI] https://www.w3.org/2019/wot/json-schema#const Provides a constant value for defined datapoint. The data type should be identical to the one as provided by the Property type.	[string] My device name	0..1
[SML] enum	[IRI] https://www.w3.org/2019/wot/json-schema#enum Provides a list of restricted set of values that the asset can provide as datapoint value.	[list of Properties<string>] enum[‘On’, ‘Off’, ‘Error’]	0..1
[Property] default	[IRI] https://www.w3.org/2019/wot/json-schema#default Provides a default value that must of the type as the datapoint valueType. The data type should be identical to the one as provided by the Property type.	[boolean] true	0..1
[Property] unit	[IRI] https://schema.org/unitCode Provides information about the datapoint’s unit. It is recommended that the unit value is assigned with a valueId from known	[string] degree:celcius	0..1
[Range] min_max	[IRI] https://admin-shell.io/idta/AssetInterfacesDescription/1/0/minMaxRange supplementalSemandicId: [IRI] (only if minimum is used) https://www.w3.org/2019/wot/json-schema#minimum [IRI] (only if maximum is used) https://www.w3.org/2019/wot/json-schema#maximum Specifies a minimum and/or maximum numeric value for the datapoint. This term is only used when type element is number or integer. When it is number, the range data type has to be float and when it is integer, the range data type has to be integer	[integer] or [float] 12..56 or 0..9.99	0..1
[Range] lengthRange	[IRI] https://admin-shell.io/idta/AssetInterfacesDescription/1/0/lengthRange supplimentalSemanticId: [IRI] (only if minimum is used) https://www.w3.org/2019/wot/json-schema#minLength [IRI] (only if maximum is used) https://www.w3.org/2019/wot/json-schema#maxLength Specifies the minimum and maximum length of a string.	[unsignedInt] 10 - 23	0..1
[SMC] items	[IRI] https://www.w3.org/2019/wot/json-schema#items Used to define the data schema characteristics (as specified within Section 2.9) of an array payload.	items |_type=integer |_min_max=0..100	0..1

[Range] itemsRange	[IRI] https://admin-shell.io/idta/AssetInterfacesDescription/1/0/itemsRange supplimentalSemanticId: [IRI] (only if minimum is used) https://www.w3.org/2019/wot/json-schema#minItems [IRI] (only if maximum is used) https://www.w3.org/2019/wot/json-schema#maxItems Defines the minimum and maximum number of items that have to be in an array payload.	[unsignedInt] 4 - 10	0..1
[Ref] valueSemantics	[IRI] https://admin-shell.io/idta/AssetInterfacesDescription/1/0/valueSemantics Provides additional semantic information of the value that is read/subscribed at runtime.	[Ref] à conceptDescription	0..1
[SMC] properties	[IRI] https://www.w3.org/2019/wot/json-schema#properties Nested definitions of a datapoint. Only applicable if type=object.	See section 2.10 properties |_timestamp |_type=string |_format=date-time |_temperature |_type=number |_min_max=-20..47 |_unit=°C	0..1

2.10 Elements of Nested SMC “properties“

Please note that the following definition is almost identical to the SMC definition for "Properties" in Section 2.8, except for the semanticId and its context.

Table 9: Element of Nested SMC properties

idShort:	properties		
Class:	SubmodelElementCollection (SMC)		
semanticId:	[IRI] https://www.w3.org/2019/wot/json-schema#properties		
Parent:	Submodel element collection with idShort = {property_name} and respective semanticId.		
Explanation:	This SubmodelElementCollection collects the nested data definition of a complex-based datapoint.		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[SMC] {property_name}	[IRI] https://www.w3.org/2019/wot/json-schema#propertyName Defines a data element within an object-based datapoint.	See Section 2.11 • [idShort] timestemp • [idShort] temperature	1..*

2.11 Elements of Nested SMC “{property_name}“

Please note that the following definition is almost identical to the SMC definition for “{property_name}“ in Section 2.9, except for the semanticId and the absent of the forms SMC and observable property.

Table 10: Elements of Nested SMC {property_name}

idShort:	{property_name} Note: {property_name} is an abstract name (e.g., “rotation_speed”) for an interaction property that includes a specific datapoint of an asset. {property_name} will detail the specifics of the datapoint (e.g., data type, restrictions, and semantics).		
Class:	SubmodelElementCollection (SMC)		
semanticId:	[IRI] https://www.w3.org/2019/wot/json-schema#propertyName		
Parent:	Submodel element collection with idShort = properties and respective semanticId.		
Explanation:	This SubmodelElementCollection defines characteristics of a datapoint element (e.g., data type, restrictions, and semantics).		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[Property] key	[IRI] https://admin-shell.io/idta/AssetInterfacesDescription/1/0/key Optional element when the idShort of {property_name} cannot be used to reflect the desired property name due to the idShort restrictions (e.g., payload message uses “temperature-value” as key term).	[string] temperature-value	0..1
[Property] title	[IRI] https://www.w3.org/2019/wot/td#title Provides a human-readable title (e.g., display a text for UI representation)	[string] Festo_Robot1	0..1
[Property] type	[IRI] https://www.w3.org/1999/02/22-rdf-syntax-ns#type Indicates the abstract data type (one of object, array, string, number, integer, boolean, or null) of the described datapoint.	[string] integer	0..1
[Property] const	[IRI] https://www.w3.org/2019/wot/json-schema#const Provides a constant value for defined datapoint. The data type should be identical to the one as provided by the Property type.	[string] My device name	0..1
[SML] enum	[IRI] https://www.w3.org/2019/wot/json-schema#enum Provides a list of restricted set of values that the asset can provide as datapoint value.	[list of Properties<string>] enum[‘On’, ‘Off’, ‘Error’]	0..1
[Property] default	[IRI] https://www.w3.org/2019/wot/json-schema#default Provides a default value that must of the type as the datapoint valueType. The data type should be identical to the one as provided by the Property type.	[boolean] true	0..1

[Property] unit	[IRI] https://schema.org/unitCode Provides information about the datapoint's unit. It is recommended that the unit value is assigned with a valueId from known unit namespaces/ontologies.	[string] degree:celcius	0..1
[Range] min_max	[IRI] https://admin-shell.io/idta/AssetInterfacesDescription/1/0/minMaxRange supplementalSemandicId: [IRI] (only if minimum is used) https://www.w3.org/2019/wot/json-schema#minimum [IRI] (only if maximum is used) https://www.w3.org/2019/wot/json-schema#maximum Specifies a minimum and/or maximum numeric value for the datapoint. This term is only used when type element is number or integer. When it is number, the range data type has to be float and when it is integer, the range data type has to be integer	[integer] or [float] 12..56 or 0..9.99	0..1
[Range] lengthRange	[IRI] https://admin-shell.io/idta/AssetInterfacesDescription/1/0/lengthRange supplimentalSemanticId: [IRI] (only if minimum is used) https://www.w3.org/2019/wot/json-schema#minLength [IRI] (only if maximum is used) https://www.w3.org/2019/wot/json-schema#maxLength Specifies the minimum and maximum length of a string.	[unsignedInt] 10 - 23	0..1
[SMC] items	[IRI]https://www.w3.org/2019/wot/json-schema#items Used to define the data schema characteristics of an array payload.	items |_type=integer |_min_max=0..100	0..1
[Range] itemsRange	[IRI] https://admin-shell.io/idta/AssetInterfacesDescription/1/0/itemsRange supplimentalSemanticId: [IRI] (only if minimum is used) https://www.w3.org/2019/wot/json-schema#minItems [IRI] (only if maximum is used) https://www.w3.org/2019/wot/json-schema#maxItems Defines the minimum and maximum number of items that have to be in an array payload.	[unsignedInt] 4 - 10	0..1

properties	[IRI] https://www.w3.org/2019/wot/json-schema#properties Nested definitions of a datapoint. Only applicable if type=object.	See section 2.10 properties |_timestamp |_type=string |_format=date-time |_temperature |_type=number |_min_max=-20..47 |_unit=°C	0..1
[Ref] valueSemantics	[IRI] https://admin-shell.io/idta/AssetInterfacesDescription/1/0/valueSemantics Provides additional semantic information of the value that is read/subscribed at runtime.	[Ref] à conceptDescription	0..1

2.12 Elements of SMC “forms”

The forms SMC specify the address information of an interaction property affordance with its datapoint. The table below shows the definition of terms that are present across all protocols.

Note: Other elements of the forms are inherited from Section 2.13.1, Section 2.13.2 and Section 2.13.3 depending on the protocol considered (HTTP, Modbus, or MQTT).

Table 11: Elements of SMC forms

idShort:	forms		
Class:	SubmodelElementCollection (SMC)		
semanticId:	[IRI] https://www.w3.org/2019/wot/td#hasForm		
Parent:	Submodel element collection with idShort = {property name} and respective semanticId.		
Explanation:	This SubmodelElementCollection defines asset datapoint endpoint resource.		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[property] contentType	[IRI] https://www.w3.org/2019/wot/hypermedia#forContentType Indicates the datapoint media type specified by IANA. Note: this local definition overwrites the globally defined contentType specified in EndpointMetadata (if it exists).	[string] application/json	0..1
[Property] href	[IRI] https://www.w3.org/2019/wot/hypermedia#hasTarget. Target IRI relative path or full IRI of asset’s datapoint. The relative endpoint definition in href is always relative to base defined in EndpointMetadata. E.g., if the base in EndpointMetadata provides “http://example.com” and the local href has “/datapoint1” as value. The full datapoint address would be “http://example.com/datapoint1”.	[string] /properties/voltage [string] http://127.0.0.1/mydata [string] sensor/temperature	1

	The specific addressing pattern for the Modbus, MQTT, and HTTP is explained in Annex B.2.	[string] 40001?quantity=2	
[Property] subprotocol	[IRI] https://www.w3.org/2019/wot/hypermedia#forSubProtocol Indicates the exact mechanism by which an interaction will be accomplished for a given protocol when there are multiple options.	[string] longpoll, websub or sse	0..1
[SML] security	[IRI] https://www.w3.org/2019/wot/td#hasSecurityConfiguration Selects one or more of the security scheme(s) that can be applied at runtime from the collection of security schemes defines in securityDefinitions SMC.	See Section 2.14 security[Ref to basic_sc]	0..1

Depending on the protocols being used, the forms SMC can be extended with protocol specific terms as explained in the following section.

2.13 Communication Protocol Bindings

Communication Bindings defines the necessary information needed to reach a datapoint via a specific protocol. These information are described in the generic forms SMC (see Section 2.12) of the interaction property “{property_name}“ SMC (see Section 2.9).
It is expected that as AID grows, some elements will overlap in many protocols. They can mean almost the same, or can mean completely different. In order to avoid misinterpretation, each specific protocol will be provided with a kind of namespace prefix tag to distinguish it. For example, for the HTTP protocol, the namespace tag htv_{parameter name} is used as a pattern. For Modbus *modv_{parameter},* for MQTT *mqv_{parameter}* and so on. The specific prefix tag is always introduced in the corresponding protocol binding section.

For each protocol, the following description question is used to identify their binding vocabularies.

1. What information is needed to reach the datapoint (data addressing information).
2. What protocol operation has to be performed (e.g., read or subscribe, write)
3. How is the content of the datapoint serialized (e.g., json, xml, octet-streams).
4. What is the logical structure of the payload including the used data types and restrictions (min, max, enums, etc)?
5. When provided, the semantical meaning of the interaction affordance and its datapoint including unit and/or context relation (e.g., to ECLASS).
6. Is there additional security parameter (apart from the one defined in EndpointMetadata) to have to be followed to access a specific datapoint.

Element names (idShorts) are chosen according to the standard names used in the protocol specification.

2.13.1 Elements of HTTP Binding for SMC forms

The following table defines all possible elements necessary for HTTP bindings within the SMC forms.

Table 12: Elements of HTTP specific terms in SMC forms

idShort:	-		
Class:	-		
semanticId:	-		
Parent:	-		
Explanation:	Extension of the SMC forms in Section 2.12 with HTTP-specific elements.		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[Property] htv_methodName	[IRI]https://www.w3.org/2011/http#methodName Defines the action to be performed datapoint IRI	[string] GET	1
[SML] htv_headers	[IRI]https://www.w3.org/2011/http#headers Defines additional information to be sent within the HTTP header message.	See Section 2.13.1.1 htv_header[{ htv_fieldName=Accept-Charset, htv_fieldValue= utf-8 }]	0..1

2.13.1.1 Elements of SML htv_headers

Table 13: Element of SML htv_headers

idShort:	htv_headers		
Class:	SubmodelElementList (SML)		
semanticId:	[IRI] https://www.w3.org/2011/http#headers		
Parent:	forms SMC		
Explanation:	This SML holds the information for http message headers definition as a SMC.		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[SMC] <no idShort>	Defines message header content	See section 2.13.2	1..*

2.13.1.2 Elements of SMC <no idShort> from SML htv_headers

Table 14: Elements of SMC <no_idShort> from SML htv_headers

idShort:	<no_idShort> **Note: according to AAS V3 Constraint AASd-120: , the Idshort of SME being a direct child of a SML shall not be specified.**
Class:	SubmodelElementCollection (SMC)
semanticId:	[IRI] https://www.w3.org/2011/http#headers
Parent:	htv_headers SML
Explanation:	This SMC holds the information for http message header definition as a SMC.

[SME type]	semanticId = [idType]value	[valueType]	card.
idShort	**Description@en**	**example**	
[Property] htv_fieldName	[IRI] https://www.w3.org/2011/http#fieldName Defines message header name	• [string] Accept-Charset • [string] Content-Length	1
[Property] htv_fieldValue	[IRI] https://www.w3.org/2011/http#fieldValue Defines message header value	• [string] utf-8 • [string] 56	1

2.13.2 Elements of Modbus binding for SMC forms

Modbus communication protocol is one of the common protocols used in industrial environment. Traditionally, it has three types. Modbus RTU, Modbus ASCII, and Modbus TCP/IP. Both modbus RTU and modbus ASCII are implemented with serial communication over RS232 or RS485 while Modbus TCP/IP is implemented over IP.

For this version of AID, Modbus TCP/IP scheme is considered because of its capability of supporting communication over Internet Protocol (IP). In future specifications, an exploration of serial communication protocols might be considered.

The following table defines all possible elements necessary for Modbus binding, they are described as Submodel elements in the InteractionMetadata Submodel element collection.

Table 15: Elements of modbus specific terms in SMC forms

idShort:	-
Class:	-
semanticId:	-
Parent:	-
Explanation:	Extension of the SMC forms in Section 2.12 with Modbus-specific elements.

[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[Property] modv_function	[IRI] https://www.w3.org/2019/wot/modbus#hasFunction Abstraction of the Modbus function code sent during a request. A function value can be either *readCoil, readDeviceIdentification, readDiscreteInput, readHoldingRegisters, readInputRegisters, writeMultipleCoils, writeMultipleHoldingRegisters, writeSingleCoil, or writeSingleHoldingRegister*	• [string] readCoil • [string] readHoldingRegisters	0..1
[Property] modv_entity	[IRI] https://www.w3.org/2019/wot/modbus#hasEntity A registry type to let the runtime automatically detect the right function code. An entity value can be *Coil, DiscreteInput, HoldingRegister,* or *InputRegister*	• [string] Coil • [string] HoldingRegisters	0..1
[Property] modv_zeroBased Addressing	[IRI] https://www.w3.org/2019/wot/modbus#hasZeroBasedAddressingFlag Modbus implementations can differ in the way addressing works, as the first coil/register can be either referred to as True or False.	[boolean] true	0..1
[Property] modv_pollingTime	[IRI] https://www.w3.org/2019/wot/modbus#hasPollingTime Modbus TCP maximum polling rate. The Modbus specification does not define a maximum or minimum allowed polling rate, however specific implementations might introduce such limits. Defined as integer of milliseconds.	[integer] 5	0..1
[Property] modv_timeout	IRI] https://www.w3.org/2019/wot/modbus#hasTimeout Modbus response maximum waiting time. Defines how much time in milliseconds the runtime should wait until it receives a reply from the device.	[integer] 5	0..1

[Property] modv_type	[IRI] https://www.w3.org/2019/wot/modbus#hasPayloadDataType Defines the data type of the modbus asset payload. type in terms of possible sign, base type. the modv_type offers a set a types defined in XML schema defined in [12]. The set of supported types value are as follows: xsd:float, xs:short ,xs:unsignedInt,,xs:string, xs:byte, xs:int, xs:boolean, xs:integer,xs:double, xs:hexbinary, xs:decimal, xs:long, xs:unsignedbyte, xs:unsignedshort, xs:unsignedint, xs:unsignedlong,	• [string] xs:float • [string] xs:unsignedInt • [string] xs:string	0..1
[Property] modv_mostSignificantByte	[IRI] https://www.w3.org/2019/wot/modbus#hasMostSignificantByte When modv_mostSignificantByte is true, it describes that the byte order of the data in the Modbus message is the most significant byte first (i.e., Big-Endian). When false, it describes the least significant byte first (i.e., Little-Endian). Note: This modv_mostSignificantByte definition will overwrite the global definition in EndpointMetadata (if it exists).	[boolean] true	0..1
[Property] modv_mostSignificantWord	[IRI] https://www.w3.org/2019/wot/modbus#hasMostSignificantWord When modv_mostSignificantWord is true, it describes that the word order of the data in the Modbus message is the most significant word first (i.e., no word swapping). When false, it describes the least significant word first (i.e. word swapping) Note: This modv_mostSignificantWord definition will overwrite the global definition in EndpointMetadata (if it exists).	[boolean] true	0..1

2.13.3 Elements of MQTT binding for SMC forms

Unlike HTTP and Modbus, MQTT uses a broker architecture that is based on publish and subscribe model. Where Publishers publish messages to specific topics and subscribers can subscribe to the topics to receive up to date notifications that matches those topics.

The following table defines all possible elements necessary for MQTT binding within the forms SMC.

Table 16: Elements of MQTT specific terms in SMC forms

idShort:	-
Class:	-
semanticId:	-
Parent:	-
Explanation:	Extension of the SMC forms in Section 2.12 with MQTT-specific elements.

[SME type]	semanticId = [idType]value	[valueType]	card.
idShort	**Description@en**	**example**	
[Property] mqv_retain	[IRI] https://www.w3.org/2019/wot/mqtt#hasRetainFlag It is an indicator that tells the broker to always retain last published payload.	[boolean] 1 or 0, true or false	0..1
[Property] mqv_controlPacket	[IRI] https://www.w3.org/2019/wot/mqtt#ControlPacket Defines the method associated to the datapoint in relation to the broker.	[string] one of “subscribe”, “publish” and “unsubscribe”	0..1
[Property] mqv_qos	[IRI] https://www.w3.org/2019/wot/mqtt#hasQoSFlag Defined the level of guarantee for message delivery between clients. 0 = atMostOnce 1 = atLeastOnce 2 = exactlyOnce	[string] default = 0 one of 0,1 and 2	0..1

It is recommended to always set the observable element in Section 2.9 to true to express that this interaction property is observ-/subscribable by the MQTT protocol.

2.14 Element of the SML security in EndpointMetadata and forms

Table 17: Element of SML security

idShort:	security
Class:	SubmodelList (SML)
semanticId:	[IRI] https://www.w3.org/2019/wot/td#hasSecurityConfiguration
Parent:	SMC EndpointMetadata or SMC forms
Explanation:	Specifies one or more security scheme that are applied for all interactions (when defined in SMC EndpointMetadata) or is valid for a specific property interaction affordance (when defined in SMC forms).

[SME type]	semanticId = [idType]value	[valueType]	card.
idShort	**Description@en**	**example**	
[Ref] <no idShort>	ReferenceElement within the SML points to a sercurity scheme definition in the SMC securityDefinitions.	security [Ref to oauth2_sc in securityDefinitions]	1..*

2.15 Elements of the SMC securityDefinitions

This section defines some well-established security mechanisms that are widely supported by protocols considered in Subsection 2.13. The securityDefinitions element contains information that could allow an asset provide access to an AAS' connection request. This access focus on the connection between asset and AAS and it is different from the AAS security itself.

For this version of AID, security definitions that are specific to HTTP and MQTT are discussed. The definitions are already available in WoT TD specification. Subsequent version of AID with additional protocols might lead to extension of the schemes that would be discussed.

Table 18: Element of SMC securityDefinitions

idShort:	securityDefinitions		
Class:	SubmodelElementCollection (SMC)		
semanticId:	[IRI] https://www.w3.org/2019/wot/td#definesSecurityScheme		
Parent:	endpointMetadata		
Explanation:	This SubmodelElementCollection holds the information about security mechanism used to access the asset.		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[SMC] {SecurityScheme}	[IRI] A collection that holds the definition of one or more security mechanisms supported by AID.	See Section 2.16	1..*

The name provided as the SecurityScheme can be arbitrary since the "scheme" element in Section 2.16 will define what kind of scheme it is. But as a good practice, it is encouraged to use the same name of the scheme as the value of the SecurityScheme SMC. Figure 3 shows the overall structure of the AID security scheme.

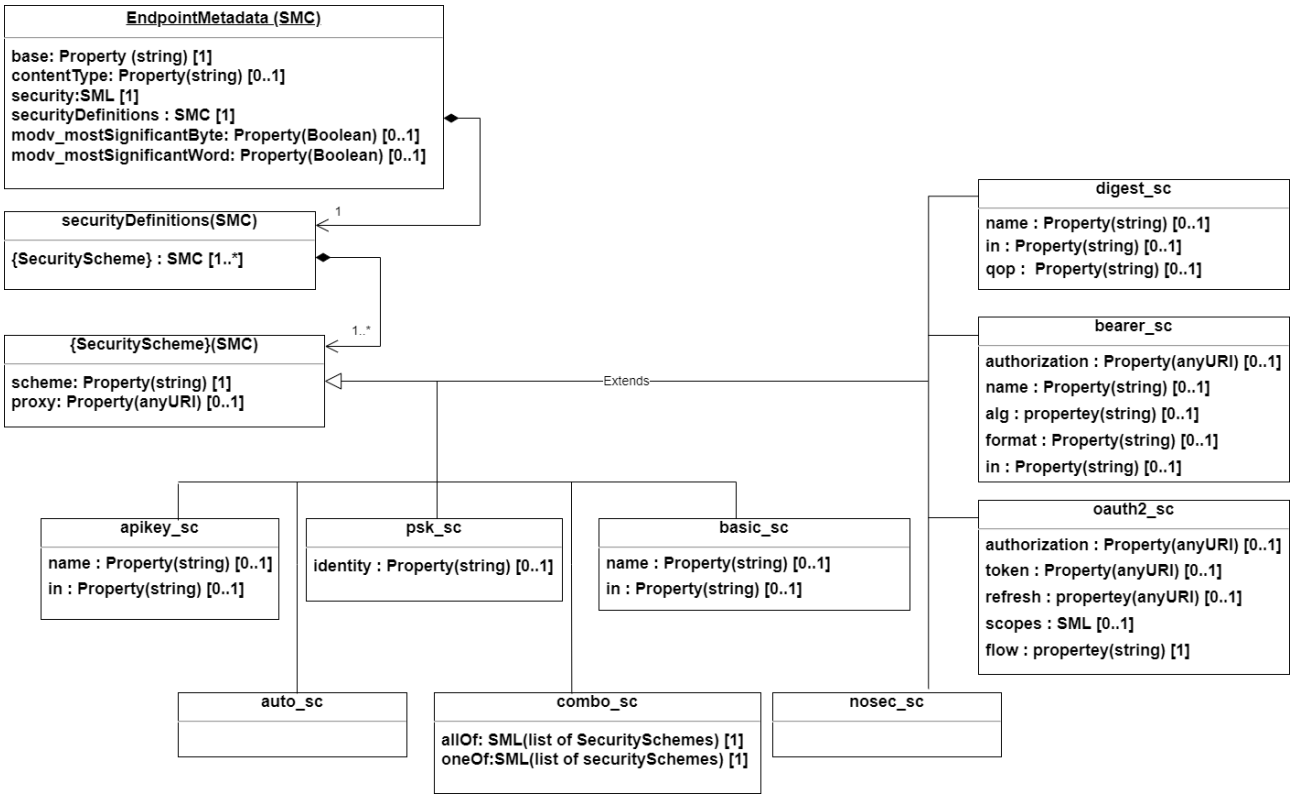


Figure 3: Overall security definitions with schemes.

2.16 Elements of the SMC {SecurityScheme}

Table 19: Elements of SMC {SecurityScheme}

idShort:	{SecurityScheme} = nosec_sc | basic_sc | digest_sc | bearer_sc | psk_sc | oauth2_sc | apikey_sc | auto_sc
Class:	SubmodelElementCollection (SMC)
semanticId:	See specific security scheme definition in the following sub-sections.
Parent:	securityDefinitions
Explanation:	This SubmodelElementCollection holds the information about security mechanism used to access the asset.

[SME type]	semanticId = [idType]value	[valueType]	card.
idShort	**Description@en**	**example**	
[Property] proxy	[IRI] https://www.w3.org/2019/wot/security#proxy Provides address information of the proxy server the security configuration provides access to.	[anyURI] "http://136.243.47.220:3128/"	0..1
[Property] scheme	[IRI] https://www.w3.org/2019/wot/security#SecurityScheme Defines the security mechanism that used during access. Supported modes one of nosec, basic, digest, bearer, psk, oauth2, apikey or auto	[string] apikey	1

As seen in Figure 3, the extension of the security scheme depends on the type of security mechanism used, the following are the list of security mechanisms presently defined AID.

- BasicSecurityScheme (basic_sc)
- DigestSecurityScheme (digest_sc)
- APIKeySecurityScheme (apikey_sc)
- BearerSecurityScheme (bearer_sc)
- PSKSecurityScheme (psk_sc)
- OAuth2SecurityScheme (oauth2_sc)
- AutoSecurityScheme (auto_sc)
- NoSecurityScheme (nosec_sc)

2.16.1 Specific Elements of basic_sc / apikey_sc for SMC {SecurityScheme}

Table 20: Element specific to basic_sc or apikey_sc of SMC {SecurityScheme}

idShort:	{SecurityScheme} = basic_sc | apikey_sc
Class:	-
semanticId:	[IRI] https://www.w3.org/2019/wot/security#BasicSecurityScheme (only for basic_sc) [IRI] https://www.w3.org/2019/wot/security#APIKeySecurityScheme (only for apikey_sc)
Parent:	-
Explanation:	This SubmodelElements holds the information about security mechanism based on basic or apikey security.

[SME type]	semanticId = [idType]value	[valueType]	card.
idShort	**Description@en**	**example**	
[Property] name	[IRI] https://www.w3.org/2019/wot/security#name Name for query, header, cookie, or uri parameters	[string] adminKey	0..1
[Property] in	[IRI] https://www.w3.org/2019/wot/security#in Specifies the location of security authentication information. Proposed values are header, query, body, cookie or auto	[string] header	0..1

2.16.2 Specific Elements of psk_sc for SMC {SecurityScheme}

Table 21: Element specific to psk_sc of SMC {SecurityScheme}

idShort:	{SecurityScheme} = psk_sc
Class:	-
semanticId:	[IRI] https://www.w3.org/2019/wot/security#PSKSecurityScheme
Parent:	-
Explanation:	This SubmodelElements holds the information about security mechanism based on psk security.

[SME type]	semanticId = [idType]value	[valueType]	card.
idShort	**Description@en**	**example**	
[Property] identity	[IRI] https://www.w3.org/2019/wot/security#identity Identifier providing information which can be used for selection or confirmation.	[string] aid-app	0..1

2.16.3 Specific Elements of digest_sc for SMC {SecurityScheme}

Table 22: Elements specific to digest_sc for SMC {SecurityScheme}

idShort:	{SecurityScheme} = digest_sc		
Class:	-		
semanticId:	[IRI] https://www.w3.org/2019/wot/security#DigestSecurityScheme		
Parent:	-		
Explanation:	This SubmodelElements holds the information about security mechanism based on digest security.		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[Property] name	[IRI] https://www.w3.org/2019/wot/security#name Name for query, header, cookie, or uri parameters	[string] adminKey	0..1
[Property] in	[IRI] https://www.w3.org/2019/wot/security#in Specifies the location of security authentication information. Proposed values are header, query, body, cookie or auto	[string] header	0..1
[Property] qop	[IRI] https://www.w3.org/2019/wot/security#qop Defines Quality of protection. Values is one of auth or auth-int	[string] auth	0..1

2.16.4 Specific Elements of bearer_sc for SMC {SecurityScheme}

Table 23: Elements specific to bearer_sc for SMC {SecurityScheme}

idShort:	{SecurityScheme} = bearer_sc
Class:	-
semanticId:	[IRI] https://www.w3.org/2019/wot/security#BearerSecurityScheme
Parent:	-
Explanation:	This SubmodelElements holds the information about security mechanism based on bearer security.

[SME type]	semanticId = [idType]value	[valueType]	card.
idShort	**Description@en**	**example**	
[Property] name	[IRI] https://www.w3.org/2019/wot/security#name Name for query, header, cookie, or uri parameters	[string] key	0..1
[Property] in	[IRI] https://www.w3.org/2019/wot/security#in Specifies the location of security authentication information. Proposed values are header, query, body, cookie or auto	[string] query	0..1
[Property] authorization	[IRI] https://www.w3.org/2019/wot/security#authorization Specifies URI of the authorization server	[anyURI] http://136.243.47.220:3128/	0..1
[Property] alg	[IRI] https://www.w3.org/2019/wot/security#alg Defines Encoding, encryption, or digest algorithm (e.g. ES256, ES512-256).	[string] ES256	0..1
[Property] format	[IRI] https://www.w3.org/2019/wot/security#format Specifies format of security authentication information. Options as value are jwt, cwt, jwe or jws	[string] jwt	0..1

2.16.5 Specific Elements of oauth2_sc for SMC {SecurityScheme}

Table 24: Elements specific to oauth2_sc for SMC {SecurityScheme}

idShort:	{SecurityScheme} = oauth2_sc
Class:	-
semanticId:	[IRI] https://www.w3.org/2019/wot/security#OAuth2SecurityScheme
Parent:	-
Explanation:	This SubmodelElements holds the information about security mechanism based on oauth2 security.

[SME type]	semanticId = [idType]value	[valueType]	card.
idShort	**Description@en**	**example**	
[Property] token	[IRI] https://www.w3.org/2019/wot/security#token Specifies URI of the token server	[anyURI] "http://136.243.47.220:3128/"	0..1
[Property] refresh	[IRI] https://www.w3.org/2019/wot/security#refresh Specifies URI of the refresh server	[anyURI] "http://136.243.47.220:3128/"	0..1
[Property] authorization	[IRI] https://www.w3.org/2019/wot/security#authorization Specifies URI of the authorization server	[anyURI] "http://136.243.47.220:3128/"	0..1
[SML] scopes	[IRI] https://www.w3.org/2019/wot/security#scopes Set of authorization scope identifiers (as Property) provided as an array. These are provided in tokens returned by an authorization server and associated with forms in order to identify what resources a client may access and how.	[list of Properties<string>] scopes[“limited”, “special”]	0..1
[Property] flow	[IRI] https://www.w3.org/2019/wot/security#flow Defines authorization flow such as code or client	[string] code	1

2.16.6 Specific Elements of combo_sc for SMC {SecurityScheme}

Table 25: Elements specific to combo_sc for SMC {SecurityScheme}

idShort:	{SecurityScheme} = combo_sc		
Class:	-		
semanticId:	[IRI] https://www.w3.org/2019/wot/security#ComboSecurityScheme		
Parent:	-		
Explanation:	This SubmodelElements holds the information about security mechanism based on combo security.		
[SME type]	**semanticId = [idType]value**	**[valueType]**	**card.**
idShort	**Description@en**	**example**	
[SML] oneOf	[IRI] https://www.w3.org/2019/wot/security#oneOf Array of two or more strings identifying other named security scheme definitions, any one of which, when satisfied, will allow access. Only one may be chosen for use.	[list of Ref] oneOf[Ref to basic_sc, Ref to bearer_sc]	1
[SML] allOf	[IRI] https://www.w3.org/2019/wot/security#allOf Array of two or more strings identifying other named security scheme definitions, all of which must be satisfied for access.	[list of Ref] allOf[Ref to basic_sc, Ref to apikey_key]	1

The combo_sc SMC defines various ways in which other named security schemes defined can be combined to create a new scheme(combo_sc). **To use the combo_sc, exactly one of either oneOf or allOf SML must be included in the SecurityScheme SMC.**

Only security scheme definitions which can be used together can be combined with allOf SML.

Other SecuritySchemes like nosec_sc and auto_sc does not extend the {SecurityScheme} SMC so they do not have extended table provided for them. **Whenever they are used, it is only recommended to define the idShort of the {secrutiyScheme} as either nosec_sc or auto_sc.**

Annex A. Explanations On Used Table Formats

1. General

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

2. Tables on 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 informations 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 language code: example@EN.
- The [valueType] is only given for Properties.

Annex B. Explanation of Parameters in AID

1. base

The base property is regarded as the entry point for asset connection. Each protocol has a specific way on how the base parameter is described. For AAS type models, the content of the element {address} is typically unknown and then represented by the placeholder "{address}" in the actual base parameter value.

The table below shows how each base parameter is described.

PROTOCOL	Description example
HTTP	http(s)://{address}:{port}/ **{address}** is the IP address or DNS address of the HTTP device. **{port}** is the port of the HTTP device.
MODBUS	modbus+tcp://{address}:{port}/{unitID}/ **{address}** is the IP address or DNS address of the Modbus device. **{port}** is the port of the Modbus device. **{unitID}** is the unit ID of the Modbus device. Generally, modbus tcp devices are identified by their IP address, the unitID should allow implementation of modbus RTU/modbus TCP gateway.
MQTT	mqtt(s)://{broker address}:{port}/ **{broker address}** is the IP address or domain name of the mqtt broker that the asset will connect to. **{port}** port is the broker port. Defauls is 1883 for 8883 for non-secured and secured connection respectively.

2. href

The href element is an extension of the base element. It can provide a fully qualified URL or a relative endpoint definition that can be combined to the information provided in base element of endpointMetadata or in the case of MQTT, provide information about resource topic.

For the sake of consistency, the href is regarded as the element that provides information about a resource endpoint. It follows the RFC3986 generic syntax for URI design.

The table below shows how an href parameter is provided for the different protocols.

PROTOCOL	Description example
HTTP	properties/voltage or https://www.assetdata.com/properties/voltage For HTTP, the href describes the path url of the datapoint in concern. This can then be combined with the base element value to make complete enpoint url that can be used to request for the datapoint value.
Modbus	`{address}?quantity={?quantity}` **{address}** Specifies the register starting address of the Modbus operation. **{quantity}** Specifies the amount of either registers or coils to be read or written to. A basic example is shown below. "40089?quantity=2"

MQTT	{topic} **{topic}** MQTT topic with the following expectations: 1) There is no topic level name '.' or '..' 2) A multi-level wildcard character (#) must be URL encoded (%23) when used 3) If the topic is used as URI reference only, a starting '/' character before the first topic level name has to be replaced by the characters './/' Some basic examples are as follows: - "mqtt://mybroker:1883/my/example/topic" → MQTT topic: "my/example/topic" - "mqtt://mybroker:1883//my/example/topic" → MQTT topic: "/my/example/topic" - "my/example/topic" → MQTT topic: "my/example/topic" - ".//my/example/topic" → MQTT topic: "/my/example/topic" - "my/example/topic/%23" → MQTT topic: "my/example/topic/#" The last example applies the URL endcoding to the “#” character.

Annex C. AID in AASX Package Explorer

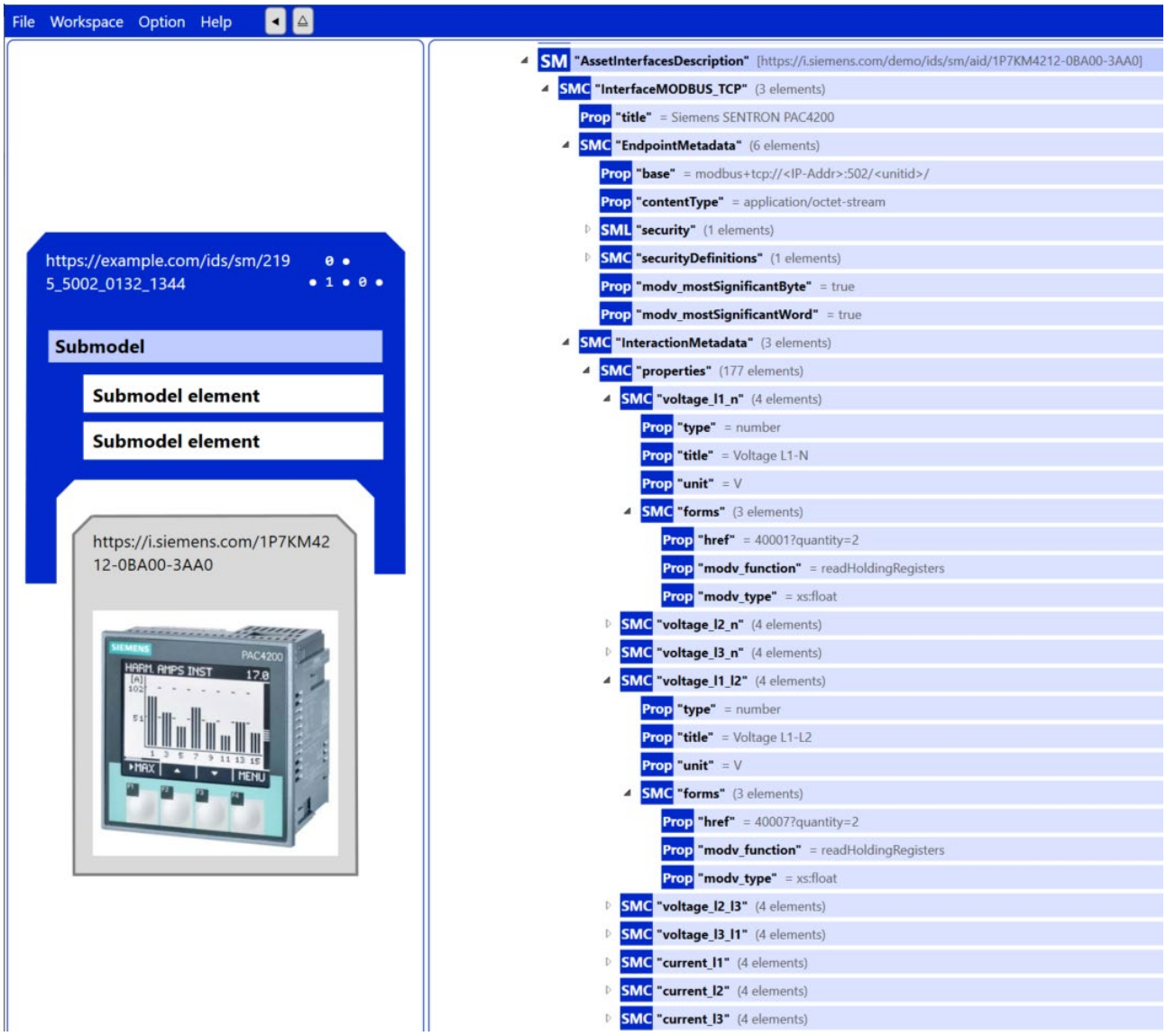


Figure 4: Example description of a device Modbus interface with its served datapoints.

Annex D. AID to WoT TD Mapping and Vice Versa

In this Annex, highlight of WoT TD terms that are covered and not covered during mapping to AID Submodel elements are provided as a table. This table is created according to the structure that WoT-TD takes (from thing class to interactionAffordance to dataSchema and securityDefinitions). The meaning of the terms provided below are already either defined in this document (see Section 2) or WoT TD document [7].

EXTERNAL = For specific term that is not covered by the AID 1.0 yet. Please use ExternalDescriptor element of the AID to refer to a corresponding Thing Description that may cover this term.

Thing Class

WoT-TD Term	Mapped AID element	Remark
@context	Is part of the used semanticId and supplementalSemanticId	This is the root namespace ID that covers all terms that will be used in AID.
@type	semanticId and supplementalSemanticId	
id	id of AID Submodel	Identifier that points to the AID Submodel
title	title as property element as string of the interface SMC	This is a human readable property element that could be used by developers for asset UI representation.
titles	EXTERNAL	If an asset “title” has multiLanguage name, the TD should be consulted for this parameter
description	The description internal object of the interface SMC	Used to provide human readable information of the interface.
descriptions	EXTERNAL	Already covered in description internal object of the interface SMC.
version	administration internal object of AID Submodel	
created	created as string property element of the interface SMC	Knowing the last an AID Submodel was created might be one of the important information for application. So created term is expected to find a place supportin AID in the future.
modified	modified as string property element of the interface SMC	
support	support as string property element of the interface SMC	
base	base property element of EndpointMetadata	This element provides entry point information URI of the asset.
properties	properties SMC of InteractionMetadata	
actions	actions SMC of InteractionMetadata	
events	events SMC of InteractionMetadata	
links	EXTERNAL	
forms	covered inside properties, actions or events SMC	
security	security SML of ReferenceElements (that points to a security scheme in securityDefinitions) inside EndpointMetadata SMC	
securityDefinitions	securityDefinitions SMC of EndpointMetata	
profile	EXTERNAL	
schemaDefinitions	EXTERNAL	
uriVariables	EXTERNAL	

form

WoT-TD Term	Mapped AID element	Remark
href	href property element of EndpointMetadata	This element provides entry point information URI of the asset.
subprotocol	subprotocol as string property element of the forms SMC	When used, it indicates interaction mechanism to used e.g “longpoll”, “websub” or “sse”.
op	EXTERNAL	For now, only read requests and subscriptions are considered so far in AID 1.0.
description	The description internal object of the property SMC	Used to provide human readable information of the property.
security	security SML of ReferenceElements inside forms SMC	
contentType	contentType as string property element of the forms SMC	

InteractionAfordances + DataSchema → properties

WoT-TD Term	Mapped AID element	Remark
observable	observable as boolean property element of the {property name} SMC	
title	title as string property element of the {property name} SMC	This is a human readable property element that could be used by developers for asset UI representation.
description	The description internal object of the {property name} SMC	Used to provide human readable information of the property.
type	type as string property element of the {property name} SMC	One of number, string, float, object, array and boolean
minimum	min_max as Range element of the {property name} SMC where minimum is assigned to min in the Range and/or maximum is assigned to max in the Range. The type of the Range should be equal to the property’s type.	Only usable for number-based values
maximum		
items	items SML of the property that covers DataSchema defintions	Only usable for array-based values
minItems	itemsRange as Range element of the {property name} SMC where minItems is assigned to min in the Range and/or maxItems is assigned to max in the Range. The type of the Range should be equal to the property’s type.	Only usable for array-based values
maxItems		
minLength	lengthRange as Range element of the {property name} SMC where minLength is assigned to min in the Range and/or maxLength is assigned to max in the Range. The type of the Range should be equal to the property’s type.	Only usable for string-based values.
maxLength		
properties (from ObjectSchema),	Nested properties SMC of {property name} SMC.	Only usable for object-based values
enum	enum as SML of the {property name} SMC containing property elements that reflects the enum entries.	
const	const as property element of the {property name} SMC.	
default	default as property element of the {property name} SMC.	

unit	unit as property element as string of the {property name} SMC.	
exclusiveMinimum, exclusiveMaximum, multipleOf, oneOf, titles, descriptions, readOnly, writeOnly, format, required, pattern, contentEncoding, contentMediaType.	EXTERNAL.	

securityDefinitions

WoT-TD Term	Mapped AID element	Remark
proxy	proxy as anyURI property element of securityDefinitions SMC	Provides URI information of the proxy server the security configuration provides access to.
scheme	scheme as SMC of security SMC	Denotes the security scheme used. It can be oneOf “nosec”, “bearer”, “basic”,”digest”, “psk”, “oauth2”, “apikey” or “auto”.
name	name as string property element of {SecurityScheme} SMC	Only usable for scheme “basic”,”digest”, “apikey”, “bearer”.
in	in as string property element of {SecurityScheme} SMC	Only usable for scheme “basic”,”digest”, “apikey”, “bearer” and value is oneOf header, query, body, cookie, or auto.
qop	qop as string property element of {SecurityScheme} SMC	Only usable for scheme “digest” and value is oneOf auth, or auth-int.
authorization	authorization as anyURI property element of {SecurityScheme} SMC	Only usable for scheme “bearer”, “oauth2”.
alg	alg as string property element of {SecurityScheme} SMC	Only usable for scheme “bearer”.
format	format as string property element of {SecurityScheme} SMC	Only usable for scheme “bearer”.
identity	identity as string property element of {SecurityScheme} SMC	Only usable for scheme “psk”.
token	token as anyURI property element of {SecurityScheme} SMC	Only usable for scheme “oauth2”.
refresh	refresh as anyURI property element of {SecurityScheme} SMC	Only usable for scheme “oauth2”.
scopes	scopes as string property element of an SML of {SecurityScheme} SMC	Only usable for scheme “oauth2”.
flow	flow as anyURI property element of {SecurityScheme} SMC	Only usable for scheme “oauth2”.

MODBUS Bindings

WoT-TD Term	Mapped AID element	Remark
modv:function	modv_function as string property of forms SMC	
modv:entity	modv_entity as string property of forms SMC	
modv:zeroBasedAddressing	modv_zeroBasedAddressing as boolean property of forms SMC	
modv:timeout	modv_timeout as integer property of forms SMC	
modv:pollingTime	modv_pollingTime as integer property of forms SMC	
modv:type	modv_type as string property of forms SMC.	• [string] xs:float • [string] xs:unsignedInt • [string] xs:string

modv:mostSiginificantByte	modv_mostSignificantByte as boolean property of endpointMetadata SMC or forms SMC	In the case that all interaction definitions use the same mostSiginificantByte value, the global definition of mostSiginificantByte in EndpointMetadata can be used instead.
modv:mostSiginificantWord	modv_mostSignificantWord as boolean property of EndpointMetadata SMC or forms SMC	In the case that all interaction definitions use the same mostSiginificantWord value, the global definition of mostSiginificantWord in EndpointMetadata can be used instead.

MQTT Binding

WoT-TD Term	Mapped AID element	Remark
mqv:retain	modbus_retain as boolean property of forms SMC	
mqv:controlPacket	modbus_controlPacket as string property of forms SMC	
mqv:qos	modbus_qos as integer property of forms SMC	

HTTP Binding

WoT-TD Term	Mapped AID element	Remark
htv:methodName	htv_methodName as string property of forms SMC	oneOf “GET”, “PUT”, “POST”,”DELETE”, “PATCH”
htv:headers	htv_headers as SMC element of forms SMC	
htv:fieldName	htv_fieldName as string property of htv_headers SMC	
htv:fieldValue	htv_fieldValue as string property of htv_headers SMC	

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] S. Kaebisch, M. McCool und E. Korkan, „Web of Things(WoT) Thing Description 1.1," World Wide Web Consortium, 2023. [Online]. Available: https://www.w3.org/TR/wot-thing-description11/

[8] Modbus Organisation, „MODBUS Messaging on TCP/IP Implementation Guide V1.0b," Modbus Organisation, 2006. [Online]. Available: https://www.modbus.org/docs/Modbus_Messaging_Implementation_Guide_V1_0b.pdf

[9] OASIS, „MQTT Version 3.1.1 Plus Errata 01," OASIS, December 2015. [Online]. Available: http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/mqtt-v3.1.1.html

[10] H. Nielsen, J. Mogul, L. M. Masinter, R. T. Fielding, J. Gettys, P. J. Leach und T. Berners-Lee, „RFC 2616 - Hypertext Transfer Protocol -- HTTP/1.1," 1999. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc2616

[11] Specification of the Asset Administration Shell; Part 2: Application Programming Interfaces, June 2023, Industrial Digital Twin Association, Available : https://industrialdigitaltwin.org/wp-content/uploads/2023/06/IDTA-01002-3-0_SpecificationAssetAdministrationShell_Part2_API_.pdf

[12] W3C XML Schema Definition Language (XSD)1.1 Part 2:Datatypes. David Peterson; Sandy Gao; Ashok Malhotra; Michael Sperberg-McQueen; Henry Thompson; Paul V. Biron et al. W3C. 5 April 2012. W3c Recommendations. Available: https://www.w3.org/TR/2012/REC-xmlschema11-2-20120405/

www.industrialdigitaltwin.org