Specification of the Asset Administration Shell

Part 4: Security

SPECIFICATION

IDTA Number: 01004
Version 3.1

This specification is part of the Asset Administration Shell Specification series.

Version

This is version 3.1 of the specification IDTA-01004.

Previous version: 3.0.2

Metamodel Versions

This document uses the following parts and versions of the “Specification of the Asset Administration Shell” series:

- IDTA-01001 Part 1: Metamodel in version 3.2 [1]
- IDTA-01002 Part 2: Application Programming Interfaces in version 3.2 [2]
- IDTA-01003-a Part 3a: Data Specification – IEC 61360 in version 3.1 [3]

If there are bugfixes of the parts, these SHALL be used.

Notice

DOI: https://doi.org/10.62628/IDTA.01004-3-1

IDTA Number: IDTA-01004

Version: 3.1

July 2026

How to Get in Contact

Contact: IDTA

Sources: GitHub

Feedback:

- new issues, bugs
- Questions and Answers

Editorial Notes

History

This document (release candidate for review) was produced from MAY 2023 to January 2025 by the Task Force “Security” of the working group "Open Technology" in the Industrial Digital Twin Association (IDTA).

Earlier content of this document was developed in the Plattform Industrie 4.0 working group “Reference Architectures, Standards and Norms“, especially Clause 7 “Security” of the Part 1 specification Version 3.0RC02 and the discussion paper “Secure Download Service” [6].

Versioning

This specification is versioned using Semantic Versioning 2.0.0 (semver) and follows the semver specification [5] with the following deviations:

- Bugfix versions are not necessarily backward compatible.

Conformance

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 RFC2119 RFC8174[1]:

- MUST word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement of the specification.
- MUST NOT This phrase, or the phrase "SHALL NOT", mean that the definition is an absolute prohibition of the specification.
- SHOULD This word, or the adjective "RECOMMENDED", mean that there MAY exist valid reasons in particular circumstances to ignore a particular item, but the full implications MUST be understood and carefully weighed before choosing a different course.
- SHOULD NOT This phrase, or the phrase "NOT RECOMMENDED" mean that there MAY exist valid reasons in particular circumstances when the particular behavior is acceptable or even useful, but the full implications SHOULD be understood and the case carefully weighed before implementing any behavior described with this label.
- MAY This word, or the adjective "OPTIONAL", mean that an item is truly OPTIONAL. One vendor MAY choose to include the item because a particular marketplace requires it or because the vendor feels that it enhances the product while another vendor MAY omit the same item. An implementation which does not include a particular option MUST be prepared to interoperate with another implementation which does include the option, though perhaps with reduced functionality. In the same vein an implementation which does include a particular option MUST be prepared to interoperate with another implementation which does not include the option (except, of course, for the feature the option provides.)

Terms and Definitions

Terms and Definitions

Please note: the definitions of terms are only valid in a certain context. This glossary applies only within the context of this document.

Please note missing terms and definitions in the review feedback form.

If available, definitions were taken from IEC 63278-1:2023 and IEC 63278-3.

Access Control

The decision to permit or deny a subject access to system objects (network, data, application, service, etc.)

[SOURCE: NIST SP 800-162]

Usage Control

Enforcement of data usage restrictions on the consumer side after access to data has been granted.

> Note 1: Usage Control is concerned with requirements that pertain to data processing (obligations) rather than data access (provisions).

[SOURCE: IDS-RAM 4 - Usage Control]

accountability

property of a system (including all of its system resources) that ensures the actions of a system entity MAY be traced uniquely to that entity, which can be held responsible for its actions

[SOURCE: IEC TS 62443-1-1:2009, 3.2.3]

authenticate

verify the identity of a user, user device, or other entity, or the integrity of data stored, transmitted, or otherwise exposed to unauthorized modification in an information system, or to establish the validity of a transmission

[SOURCE: IEC TS 62443-1-1:2009, 3.2.12]

authentication

security measure designed to establish the validity of a transmission, message, or originator, or a means of verifying an individual’s authorization to receive specific categories of information

[SOURCE: IEC TS 62443-1-1:2009, 3.2.13]

authorization

right or permission that is granted to a system entity to access a system resource

[SOURCE: IEC TS 62443-1-1:2009, 3.2.14]

data integrity

property that data has not been changed, destroyed, or lost in an unauthorized or accidental manner

> Note 1 to entry: This term deals with constancy of and confidence in data values, not with the information that the values represent or the trustworthiness of the source of the values.

[SOURCE: IEC TS 62443-1-1:2009, 3.2.38]

Abbreviations

Abbreviation	Description
AAS	Asset Administration Shell
AASX	Package file format for the AAS
ABAC	Attribute Based Access Control
ACL	Access Control List
API	Application Programming Interface
BLOB	Binary Large Object
BNF	Backus Naur Form
DKE	Deutsche Kommission für Elektrotechnik
HTTP	Hypertext Transfer Protocol
ID	Identifier
IDTA	Industrial Digital Twin Association
IDP	Identity Provider
IEC	International Electrotechnical Commission
IRDI	International Registration Data Identifier
ISO	International Organization for Standardization
JSON	JavaScript Object Notation
MIME	Multipurpose Internet Mail Extensions
OAUTH	Open Authorization
ODRL	Open Digital Rights Language
OIDC	OpenID Connect
OPC	Open Packaging Conventions (ECMA-376, ISO/IEC 29500-2)
RDF	Resource Description Framework
REST	Representational State Transfer
RFC	Request for Comment
ROA	Resource Oriented Architecture
SOA	Service Oriented Architecture

Abbreviation	Description
UML	Unified Modeling Language
URI, URL, URN	Uniform Resource Identifier, Locator, Name
VDE	Verband der Elektrotechnik Elektronik Informationstechnik e. V.
VDI	Verein Deutscher Ingenieure e.V.
VDMA	Verband Deutscher Maschinen- und Anlagenbau e.V.
W3C	World Wide Web Consortium
XACML	eXtensible Access Control Markup Language
XML	eXtensible Markup Language
X509	Standard format for public key certificates
ZIP	archive file format that supports lossless data compression
ZVEI	Zentralverband Elektrotechnik- und Elektronikindustrie e. V.

[1] https://www.ietf.org/rfc/rfc2119.txt

Preamble

Metamodel Versions

This document uses the following parts of the “Specification of the Asset Administration Shell” series:

- IDTA 01001 Part 1: Metamodel in version 3.1 [1]
- IDTA 01002 Part 2: REST API in version 3.1 [2]
- IDTA-01003-a Part 3a: Data Specification – IEC 61360 in version 3.1 [3]
- IDTA 01005 Part 5: Package File Format (AASX) in version 3.1 [4]

Scope of this Document

This document specifies the security for the Asset Administration Shell and its submodels, i.e. how to use Access Tokens and how to define Access Rules for Authorization. Identifiables (i.e. Asset Adminstration Shells, Submodels and Concept descriptions) can also be signed by the additional endpoint /$signed, as described in the REST API specification.

This document includes the grammar of a technology neutral model, which is used both for HTTP API 3.1 Query Language and for the Access Rules. In addition, a corresponding JSON schema is defined.

Structure of the Document

Clause *Terms and Definitions* lists Terms, Definitions and Abbreviations

Clause *Introduction* gives a detailed introduction to the security topic

Clause *Access Rule Model (normative)* defines the Access Rule Model (normative)

Clause *Accountability for AAS Content and Digital Signatures* describes accountability for AAS content and digital signatures

Clause *Secure Asset Binding* describes secure asset binding

Clause *Summary and Outlook* provides a summary and outlook

Annex Backus-Naur-Form defines the grammar language used in the specification

Annex Examples of Access Rules in text serialization contains Examples of Access Rules in text serialization

Annex Examples of Access Rules in JSON serialization contains Examples of Access Rules in JSON serialization

Introduction

This document explains the security of the Asset Administration Shell.

This specification defines a model for access rules. Such model targets 2 purposes.

Purpose 1 is to provide guidelines for implementations.

Purpose 2 is to address the definition and exchange of access rules.

The exchange of access rules is very useful for complex assets like robot or machine. A set of access rules is provided by the supplier of an asset and the operator adjusts the access rules to his needs.

In case of switching between AAS implementations, an exchange of access rules is useful.

AAS implementations (of non test systems) are supposed to consider relevant security standards and regulations, e.g. IEC 62443, ISO 27001, EU CRA or EU NIS2. This includes rules for the development process and for the written code.

Section Requirements defines security requirements for AAS systems following the IEC 63443 series. Implementation specific security requirements are not part of this document. The Annex Requirements Mapping provides an overview on which requirements are addressed by this document and which are not.

Typical scenarious for AAS include multiple stakeholders along the industrial value chain. This comes with the need for authentication and authorization of subjects, i.e., users and systems, accross different security domains. This is addressed by identity and access management (IAM). This document dicusses several patterns and examples for IAM in the context of AAS (Integration Patterns,External and Internal Identity Providers, Authentication Flows). However, this document does not mandate a specific solution for IAM. As a result of the subject identification and authentication process there is an Access Token that is evaluated by the AAS to authorize access to data and services represented by the AAS.

An Access Rule Model is defined, which uses Attribute Based Access Control (Attribute Based Access Control (ABAC)) as underlying concept. Claims carried by the Access Token are used as attributes for the ABAC authorization. ABAC allows to consider additional attributes which go beyond subject identity or role for authorization decisions. Date and time of the requiest, current machine state, environmental conditions are examples for additional attributes.

Access rules are defined for routes in the AAS API [2] (part 2), for Identifiables and Referables by reference or semanticId, or for certain assets.

Types of AAS

As introduced in AAS Part 2 API, Figure 1 shows 3 different types of information via the Asset Administration Shell.

- Type 1 is the exchange as an AASX file [3] (part 5). Such AASX file can include a Security Model to be passed to a business partner.
- Type 2 is the access to AAS by the API [2] (part 2) to a server. Such server can use and enforce the ABAC rules of the Security Model. The Security Model may be hosted as submodel by the server. The submodel for the Security Model is then itself subject to access control.
- Type 3 is the so called “Industrie 4.0 Language” with active AAS, which is similar to agents. Basically Type 3 is out of scope for this document, but the definitions of this document can also be used in the context of Type 3.

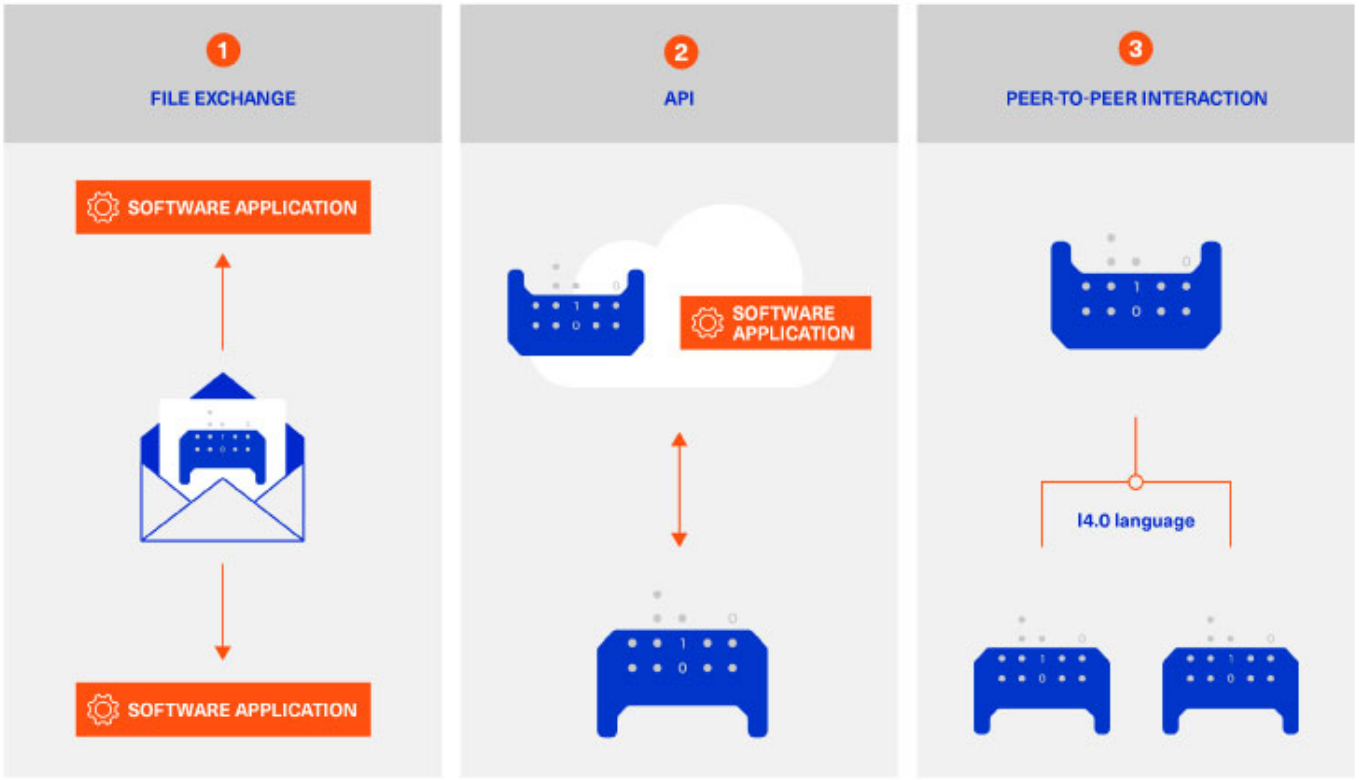


Figure 1. Types of Information Exchange via Asset Administration Shells

Services, Interfaces and Interface Operations

AAS Part 2 API also introduces the Industrie 4.0 Service Model illustrated in Figure 2 for a uniform understanding and naming. It basically distinguishes between associated concepts on several levels (from left to right):

- **technology-neutral level**: concepts that are independent of selected technologies;
- **technology-specific level**: concepts that are instantiated for a given technology and/or architectural style (e.g. HTTP/REST, OPC UA, MQTT);
- **implementation level**: concepts that are related to an implementation architecture that comprises one or more technologies (e.g. C#, C++, Java, Python);
- **runtime level**: concepts that are related to identifiable components in an operational Industry 4.0 system.

This document deals with the concepts of the technology-neutral and technology-specific level. However, to avoid terminological and conceptual misunderstandings, the whole Industrie 4.0 Service Model is provided here.

The technology-neutral level comprises the following concepts:

- **Service**: a service describes a demarcated scope of functionality (including its informational and non-functional aspects), which is offered by an entity or organization via interfaces.
- **Interface**: this is the most important concept as it is understood to be the unit of reusability across services and the unit of standardization when mapped to application programming interfaces (API) in the technology-specific level. One interface may be mapped to several APIs depending on the technology and architectural style used, e.g. HTTP/REST or OPC UA, whereby these API mappings also need to be standardized for the sake of interoperability.
- **Interface-Operation**: interface operations define interaction patterns via the specified interface.

The technology-specific level comprises the following concepts:

- **Service Specification**: specification of a service according to the notation, architectural style, and constraints of a selected technology. Among others, it comprises and refers to the list of APIs that forms this service specification. The service specification includes I4.0-defined standard APIs but also other, proprietary APIs.

Note: such a technology-specific service specification may be but does not have to be derived from the “service” described in the technology-neutral form. It is up to the system architect and service engineer to tailor the technology-specific service according to the needs of the use cases.

- **API**: specification of the set of operations and events that forms an API in a selected technology. It is derived from the interface description on the technology-neutral level. Hence, if there are several selected technologies, one interface may be mapped to several APIs.
- **API-Operation**: specification of the operations (procedures) that are accessible through an API. It is derived from the interface operation description on the technology-neutral level. When selecting technologies, one interface operation may be mapped to several API-operations; several interface operations may also be mapped to the same API-operation.

The implementation level comprises the following concepts:

- **Service-Implementation**: service realized in a selected implementation language following the specification in the Service Specification description on the technology-specific level.
- **API-Implementation**: set of operations realized in a selected implementation language following the specification in the API description on the technology-specific level.
- **API-Operation-Implementation**: concrete realization of an operation in a selected implementation language following the specification in the API-Operation description on the technology-specific level.

The runtime level comprises the following concepts:

- **Service-Instance**: instance of a Service-Implementation including its API-Instances for communication. Additionally, it has an identifier to be identifiable within a given context.
  1. **API-Instance**: instance of an API-Implementation which has an endpoint to get the information about this instance and the related operations.
  2. **API-Operation-Instance**: instance of an API-Operation-Implementation which has an endpoint to get invoked.

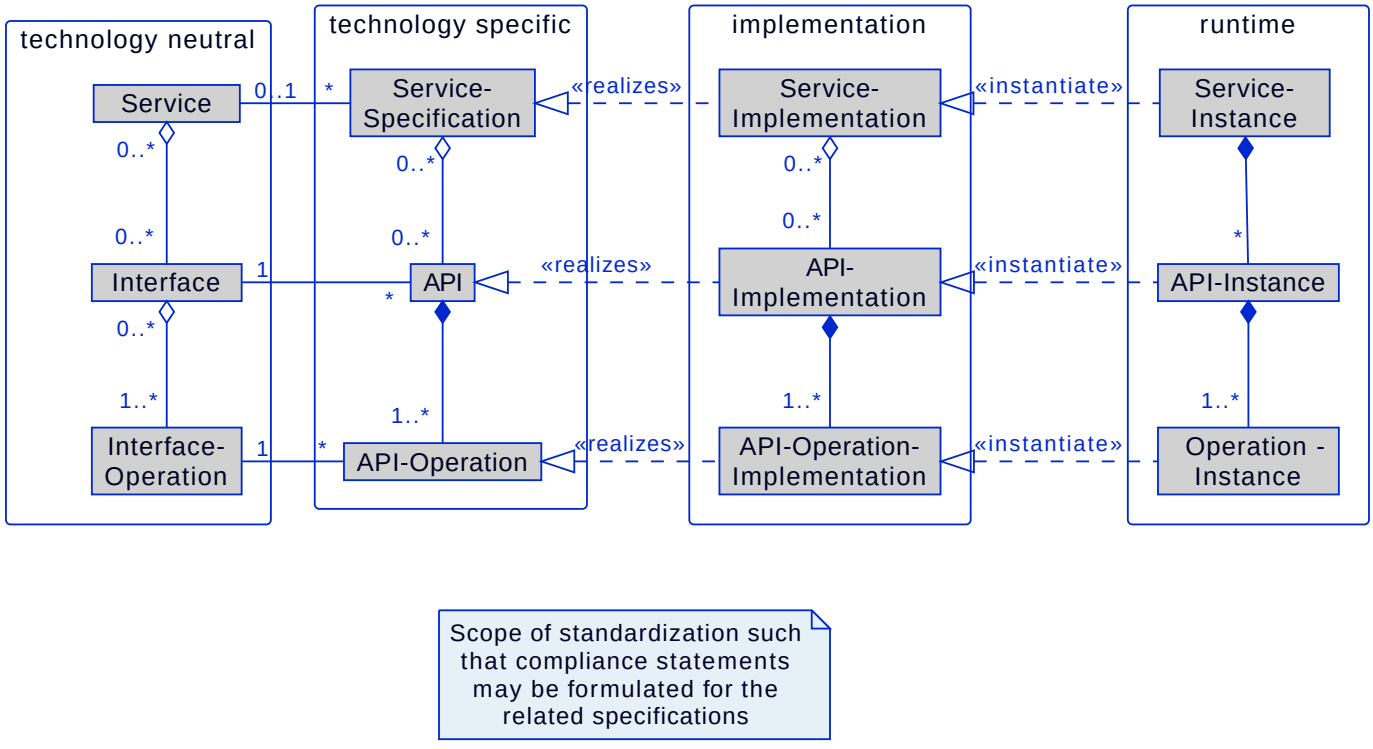


Figure 2. Services, Interfaces and Interface Operations

One important message from the Industrie 4.0 Service Model is that it is the level of the interface (mapped to technology-specific APIs) that

- provides the unit of reusability,
- is the foundation for interoperable services, and
- provides the reference unit for compliance statements.

Figure 3 shows AAS Services/Interfaces and an example sequence how they are called from a client application:

- At first a client application provides an asset ID (asset link) to the AAS Discovery Interface to retrieve the corresponding AAS ID or AAS IDs.
- By the AAS ID the related AAS Descriptor can be retrieved through the AAS Registry Interface.

An AAS descriptor includes the endpoint of the AAS and of the related Submodels.

- AAS or Submodel are either hosted standalone or as part of a larger AAS or Submodel Repository. In Figure 3 the first submodel is accessed by the Submodel Interface and the second submodel is accessed by the Submodel Repository Interface.

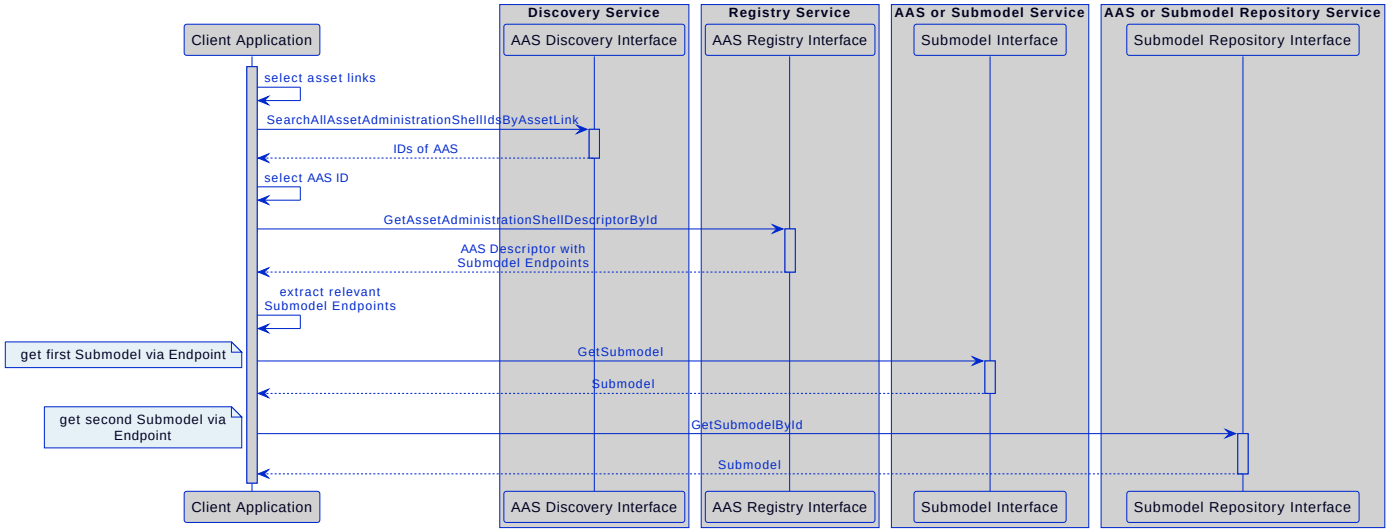


Figure 3. Sequence Diagram of AAS Services

Use Case File Exchange

Figure 4 shows an example use case of File Exchange between business partners as introduced by AAS Part 5 Package File.

In the example a supplier sends AAS as AASX by email to an integrator. The integrator sends the AAS received from the supplier and additional own AAS to his customers. It is up to the integrator to forward only selected submodels from the supplier's AAS to his customer.

This use case is an example for the necessity of Signing, since it is desireable to check the integrity of the AAS originally provided by the supplier.

Chapter Accountability for AAS Content and Digital Signatures discusses means of ensuring accountability of AAS data. Basic signing functionality for data retrieval from an AAS is defined in the REST API specification IDTA-01002.

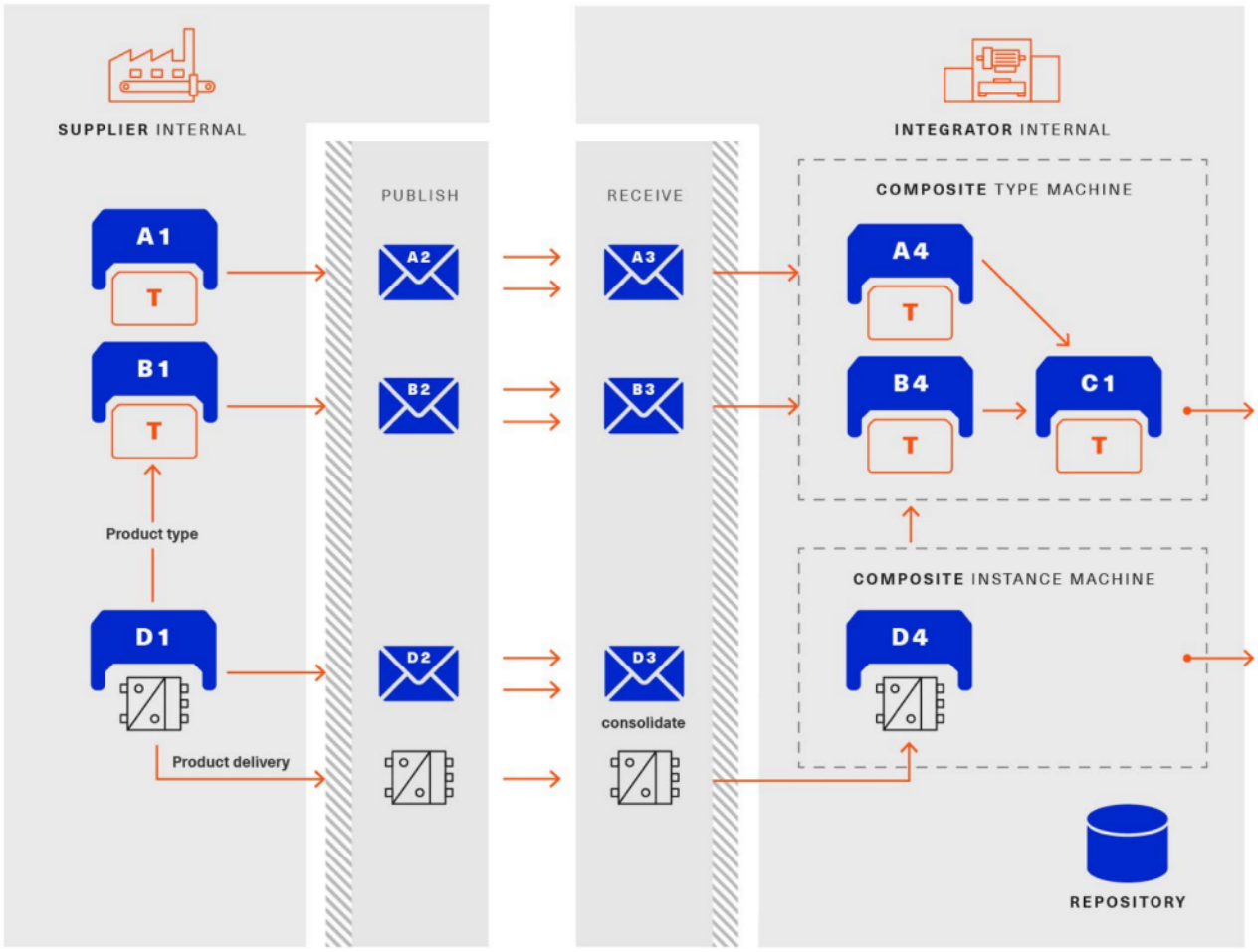


Figure 4. Use Case File Exchange between Value Chain Partners

When having a look at the Use Case File Exchange in Figure 4 also security aspects have to be considered when transferring information from one value chain partner to the next.

When AAS content is passed from one partner to another, this is typically related to a change in the access control domain of the partners involved (supplier, integrator, operator), i.e. the scope of the validity of access control policies.

Therefore, for the example that the supplier passes on data to the integrator, the following typical steps are carried out:

- Step A1-A2: The supplier makes a choice which data is to be passed on, and thus determines which APIs are accessible to whom and/or the content of the AASX package.
- Step A2-A3: the AASX package is transferred to the integrator. With API this step is only needed, if the supplier pushes (POST, PUT) the AAS data to the integrator.
- Step A3-A4: The integrator receives the AAS by the API or receives the package and imports the content into his security domain. During this step, the integrator has to establish access rights according to the requirements in his own security domain.

The exchange of access permission rules is described in Exchange of Access Rules. == Use Case AAS Servers with API

Figure 5 shows an example of a User Application accessing 3 AAS Servers and a Registry. This example will be further detailed in Architecture Examples below.

The User Application will access the registry by the Registry API to retrieve the endpoints of AAS and Submodels.

It will then access the data on the different AAS servers by the related APIs.

This use case is an example for the necessity of Authentication, Authorization and Signing.

The AAS Servers provide data available to the public, i.e. anonymous users, or restrict the access of certain data to specific users. To gain access to restricted data, the User Application first authenticates (not in the figure) and gets an Access Token to supply its identity and further attributes to an AAS Server and also to the Registry. Access rules will be validated and enforced using the attributes in the Access Token.

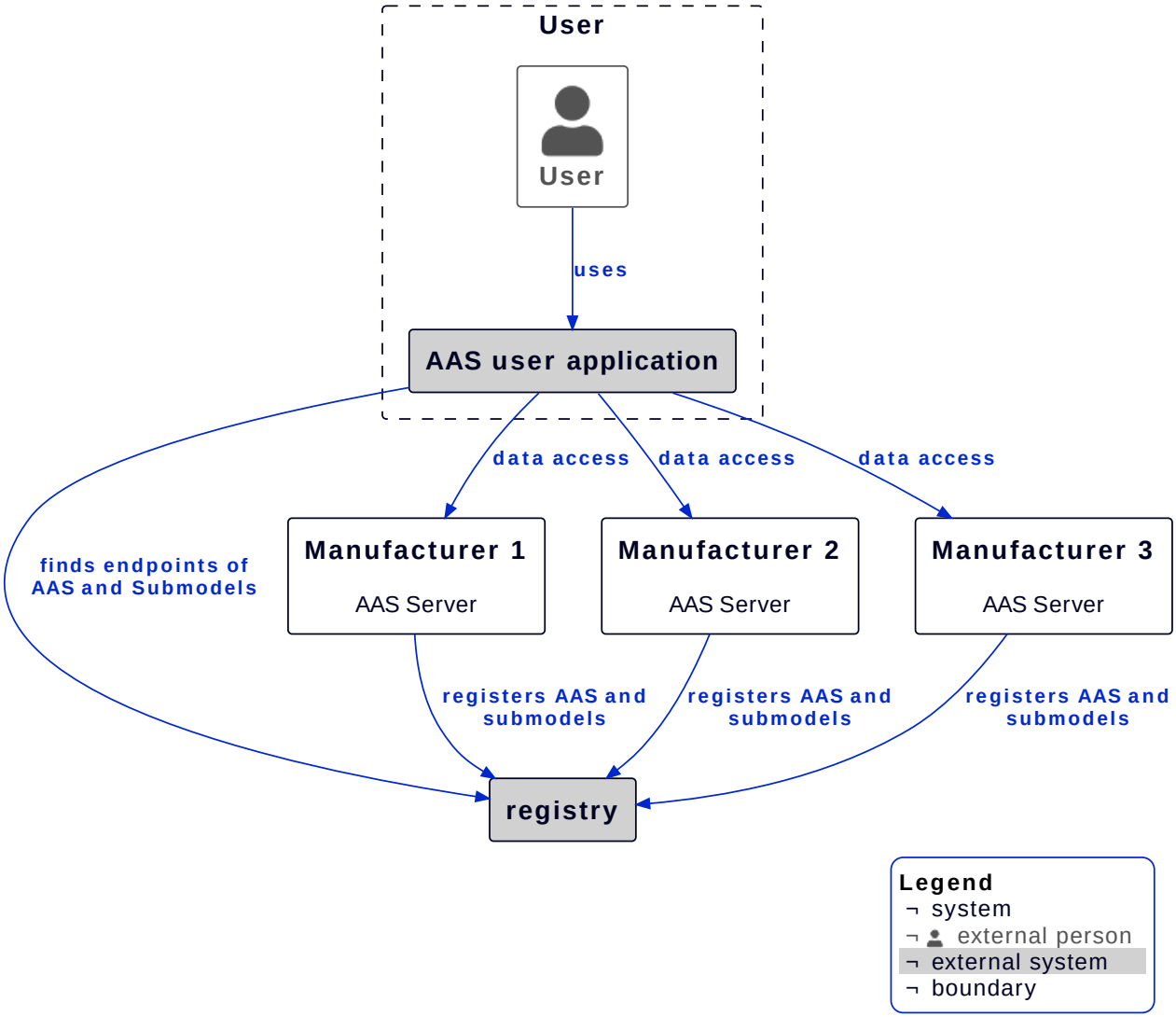


Figure 5. Example of a User Application accessing 3 AAS Servers and a Registry

AAS Server means either an AAS Repository with multiple AAS or just an AAS endpoint with a single AAS.

Further use cases are available at the Platform Industrie 4.0, e.g. Collaborative Condition Monitoring or Exchange of Engineering Data, and in IEC 63278-4 Use Cases.

Repository and registry

AAS Security applies to all AAS APIs, especially to both repositories and registries.

In case of repositories the Access Rules define the access to the data in the repository itself.

A registry only includes AAS descriptors or submodel descriptors. In that case the Access Rules define the access to the descriptors, i.e. by rules with semanticId, assetId or ID for identifiable.

This gives the possibility to copy the related rules made for a repository also to a registry, either manually or even automatically.

Architecture Examples

Protecting an AAS Server

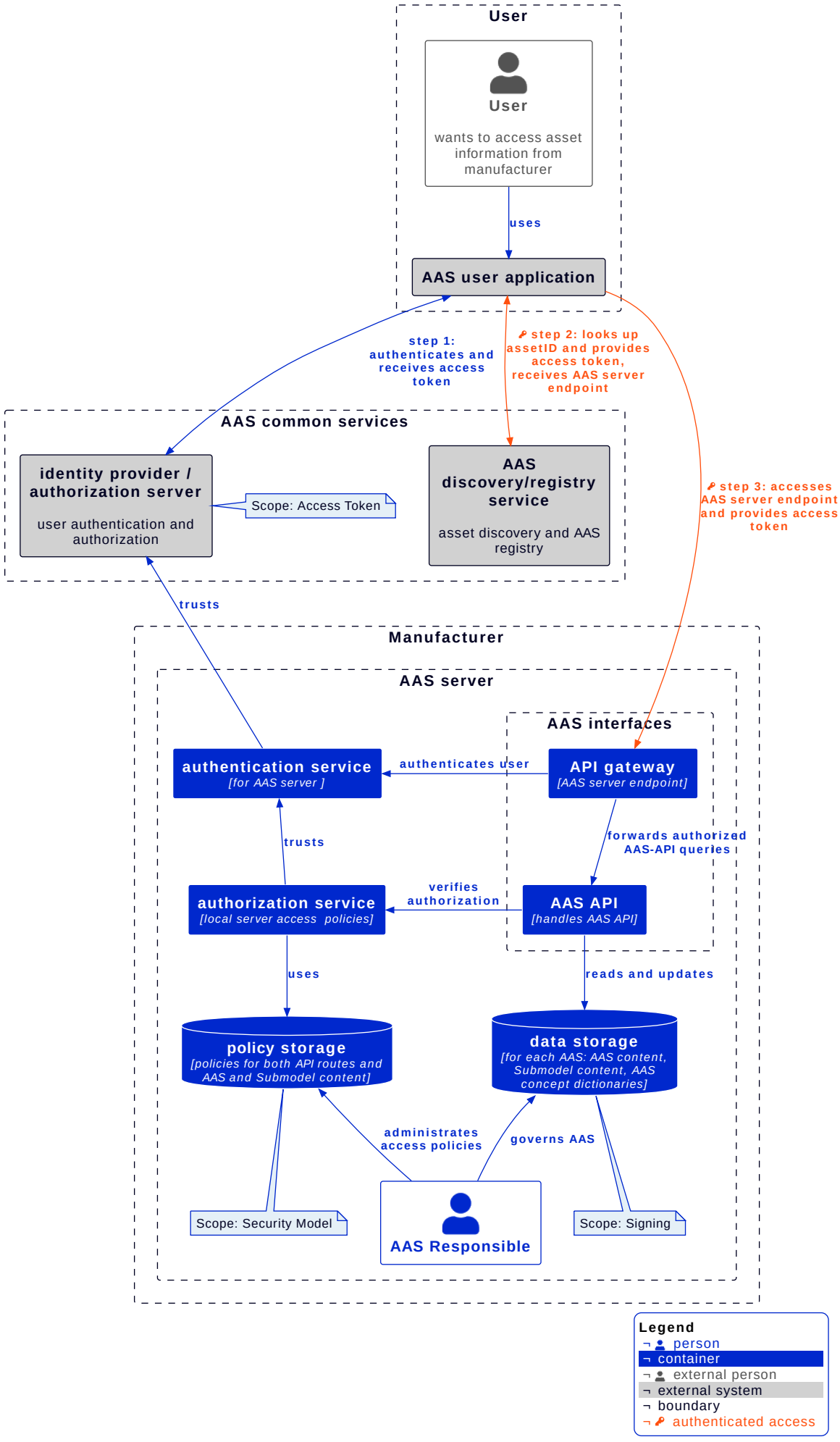


Figure 6. Example of a possible AAS server implementation

Figure 6 shows a possible AAS server implementation with security:

- A **User** wants to access AAS information from a **Manufacturer** using an **AAS User Application**.
- The **Manufacturer** hosts an **AAS Server** which is internally administrated by an **AAS responsible**.
- In step 1 the **User Application** authenticates at and receives an access token from the **Identity Provider / Authorization Server**.
- In step 2 the **User Application** looks up e.g. the aasId at the **AAS Discovery/Registry Service** and receives the related AAS endpoint for the **AAS Server**.
- In step 3 the **User Application** accesses such endpoints in the **AAS Server**.
- First such access will be handled by the **API Gateway** in the **AAS Server**. Such **API Gateway** will verify the signature of the access token by the **Authentication Service**, which uses the public key of the **Identity Provider / Authorization Server**.
- Second the **AAS API** verifies by the **Authorization Service**, that the requested API route is authorized. Access is determined by related ABAC access rules for that API route.
- Third the **AAS API** verifies by the **Authorization Service**, that the requested AAS element is authorized. Access is determined by ABAC access rules with IDs of Identifiables, Referables, Assets or semanticIds.
- Fourth the **AAS API** will access the **Data Storage**, but only in case of all the positive authorizations above.
- Business partners send proposed access rules to the **AAS Responsible**, e.g., a machine supplier proposes such rules to a machine operator. Whether such access rules are stored in a proprietary way by the **AAS Server**, or in the Data Storage in AAS format as received depends on the implementation.
- As described for the **AAS Server**, the AAS **User Application** will also provide the access token to the **Discovery/Registry Service** which can make the verification and authorization accordingly.

Protecting an AAS Registry

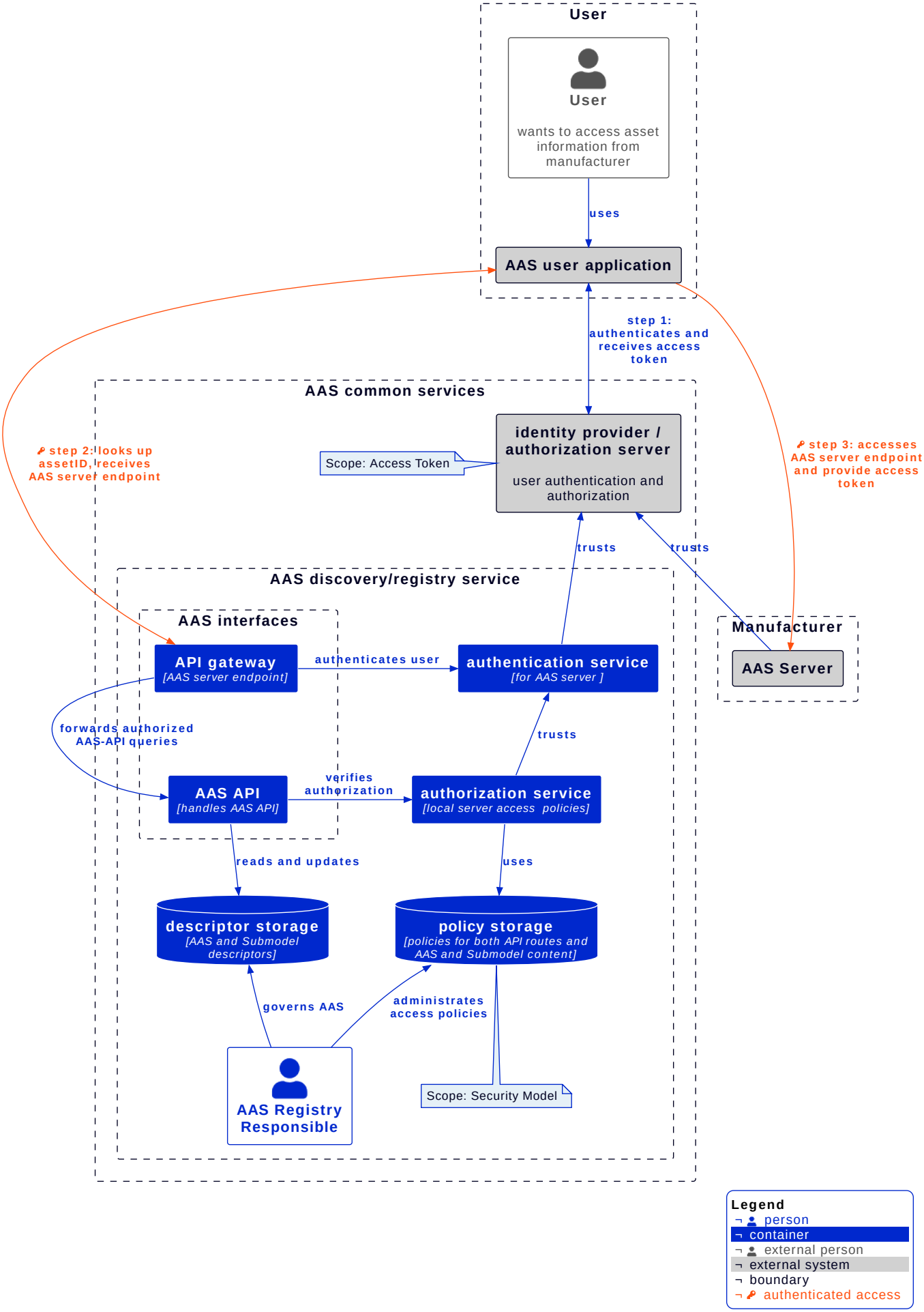


Figure 7. Example of a possible AAS registry implementation

Figure 7 shows a possible AAS registry implementation with security:

- The same example as before is shown, but this time the **Discovery/Registry Service** is shown in more detail.
- Instead of AAS Data only AAS and Submodel descriptors are stored in the descriptor storage.

Attribute Based Access Control (ABAC)

The objective of access control is to protect system resources (here: AAS content) against unauthorized access. The protection measures are specified in access control policies whose scope of validity is defined by security domains dedicated to access control.

AAS applications require effective management of data exposure. This is especially important if data is shared across multiple companies. Attribute based access control provides a means of addressing these requirements. Refer to [8] for further explanations of ABAC. Originally, ABAC relies upon the data-flow model and language model of the OASIS eXtensible Access Control Markup Language (XACML) specifications [9].

Many existing systems for access control in IACS apply role based access control (RBAC). RBAC employs pre-defined roles. Each subject (AAS user application or AAS user application user in case of AAS) is assigned one or several roles. Each role comes with a set of privileges. Before access is allowed to an AAS object (6.2.4.5) it is checked whether the role associated with the requesting subject has sufficient privileges. The role assigned to a subject represents one specific attribute of that subject. This attribute “role” is typically derived from the subjects identity (5.2.2.1). ABAC extents the traditional concept of RBAC. ABAC it allows to use additional attributes other than the subject’s role in the evaluation of an access decision. It is still possible to stick with a single attribute “role” to maintain interoperability towards an RBAC environment.

Each access decision within ABAC is taken based on a set of access control rules based on ABAC attributes:

- Subject attributes, i.e., attributes of the subject requesting AAS information, e.g., identity, role/responsibility, affiliation of AAS user application or AAS user application user
- AAS object attributes, i.e., attributes of the AAS object (6.2.4.5) being requested, e.g., semanticId of a Submodel, SubmodelElement, or Property.
- Environmental conditions/global attributes, e.g., time at which an enquiry is made.

A typical ABAC system is built around the following function blocks:

- Policy administration point (PAP): The system entity that creates a policy set.
- Policy decision point (PDP): The system entity that evaluates an applicable policy and renders an authorization decision.
- Policy enforcement point (PEP): The system entity that performs access control, by making decision requests and enforcing authorization decisions.
- Policy information point (PIP): The system entity that acts as a source of attribute values.

The general request flow is depicted in Figure 8:

- A subject is requesting access to an object (1). In the context of an AAS, an object is for example a submodel or a property or any other submodel element.
- The implemented access control mechanism of the AAS evaluates the access permission rules (2a) that include constraints that need to be fulfilled w.r.t. the subject attributes (2b), the object attributes (2c) and the environment conditions (2d).
- After the evaluation a decision is taken and enforced upon the object (3), i.e. the access to the object is permitted or declined.

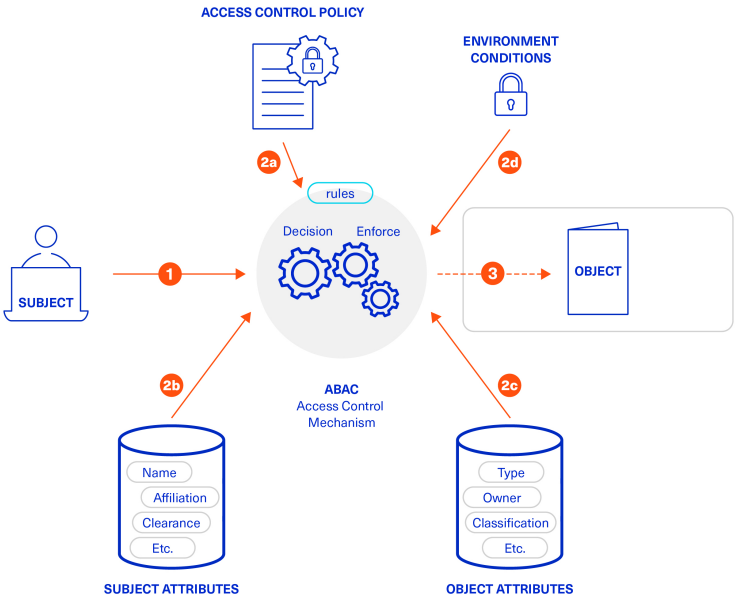


Figure 8. Attribute Based Access Control

In this document the concept of attributes is specialized for the application of ABAC in the context of AAS authorization.

Attributes are claims from an access token (e.g. subject attributes), global attributes (e.g. environmental conditions) or attributes from AAS and submodel data (e.g. object attributes).

Dataspaces

A Dataspace facilitates the exchange of data between participants via a set of technical services. Each participant is capable of implementing and operating a set of services/agents that allows them to participate in data exchange. A Dataspace Authority manages the specifications and conventions for these interactions. Identity Providers issue identity tokes to the participants who can in turn decide which issuers to trust.

AAS Part 2 defines interoperability on the level of standardized interfaces serving business objects leveraging the semantics of AAS Part 1. Neither specification makes normative statements about common mechanisms to establish authenticity, identification or authorization between consumers. This is however necessary to define the terms and conditions for secure access to the standardized AAS-interfaces.

Interoperability across all layers of a Dataspace requires as answer a layered set of specifications with well-defined integrations among them.

In a dataspace members can exchange data securely and interoperably.

Such data exchange is always done in the same way, so that scalability and efficiency can be achieved.

Millions of business partners are able to easily exchange data with each other in a dataspace.

The data exchange can be with and without user interaction.

To achieve interoperability, members of a dataspace agree on:

- Common models
- Common APIs
- Common security (e.g. credentials, authentication, authorization)

For a dataspace with AAS there are already specifications available:

- AAS metamodel
- AAS API, including AAS Discovery and AAS Registry
- Many AAS Submodel Templates (e.g. Nameplate) conformant to the AAS metamodel
- AAS Security (i.e. this document)

In addition, a dataspace with AAS needs to define:

- Further dataspace specific Submodel Templates
- Allowed concept repositories to be referenced by semanticIds
- Technology for identities, e.g. X509 or Verifiable Credentials
- Technology for identity providers, e.g. OAUTH, OIDC, Dataspace Protocol/Decentralized Claims Protocol
- Using central or decentral identity providers
- Claims used and allowed in JSON Web Tokens issued by identity providers

Remark: Until now no common AAS specifications for the following concepts are published and need also to be defined in a dataspace with AAS:

- How to find AAS registries of the dataspace members? A solutions is to have a registry with registry endpoints or a naming convention like registry.company-domain.com. Also, an indirect registration via dataspace connectors that also support additional data exchange patterns besides AAS, e.g. simple bilateral file exchange, is possible.
- How to find concept repositories? A solution can be a registry of concept repository endpoints or a concept repository discovery service for semantic IDs. Additional discovery services if needed, for example for the company-domain etc.

Integration Patterns

With AAS many use cases between business partners are supported. In this chapter different integration patters, see Figure 9, are described, which are used in such use cases and how standardized AAS Security supports this.

An integration pattern describes

- if an own AAS or an AAS of a business partner is used.
- if an AAS is hosted on a server internally or externally.
- if an AAS is copied.
- the requirements on Identity Providers.
- the requirements on access tokens and claims (as part of a JSON Web Token).
- the requirements on access rules.

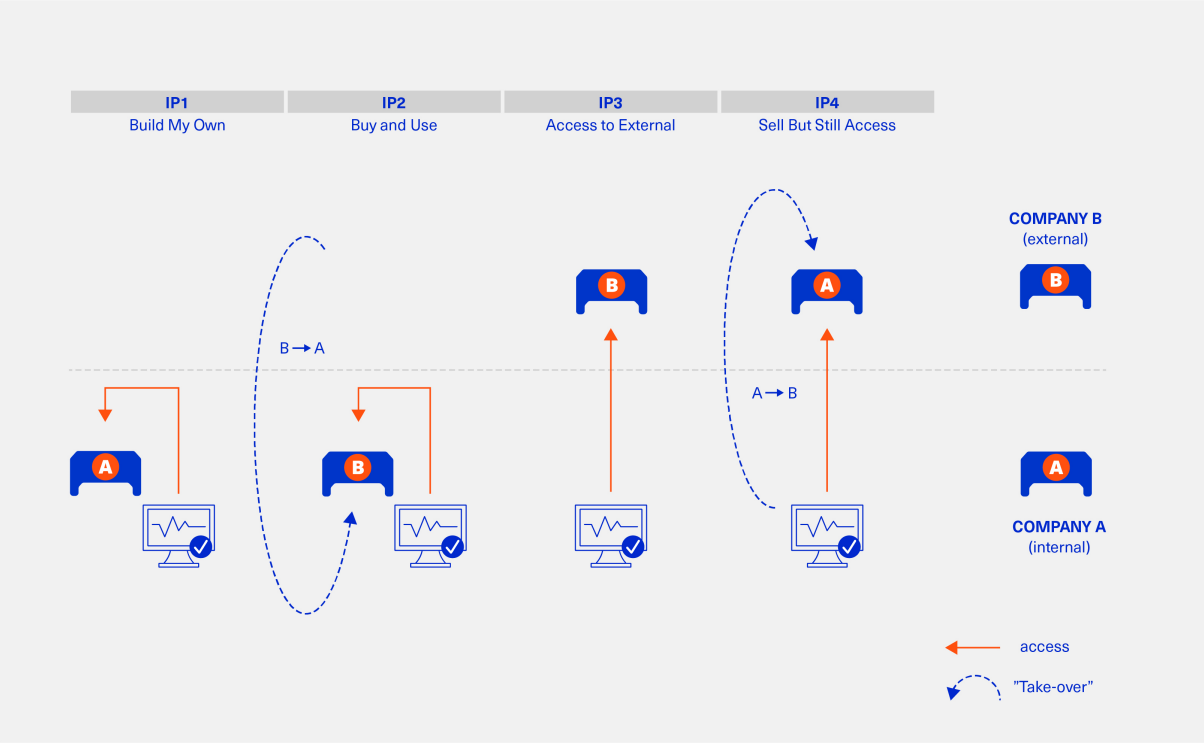


Figure 9. Integration patterns

Integration Pattern “Build My Own” (IP1)

- Company A creates, manages and uses its own AAS. The AAS is not accessible from outside the Company.
- Everything can be decided internally by the company (Company A).
- Company hosts AAS on a server internally. The server software is open source, a commercial product or a software as a service.
- Company chooses an Identity Provider, that is aligned with company’s IT infrastructure. The chosen server software is configured to use this Identity Provider.
- Chosen server software either supports a set of predefined claims or is fully configurable for claims.
- Chosen server software either supports a limited set of build-in access rules or is fully configurable for access rules according standardized AAS security.
- For many internal Use Cases the security is expected to be complete company specific and does not depend on using the standardized AAS security. However, if the company decides later to make their AAS available to other companies (pattern “Access to External”) or the company chooses to use existing software to host their AAS, the company does still benefit from using standardized AAS security.

Integration Pattern “Buy and Use” (IP2)

- Company A buys a product from Company B and uses and manages it internally.
- Company B has created the AAS and Company A copies the AAS.
- The AAS is hosted on a server of company A.
- Company A defines Identity Provider, access token and claims.
- Company B proposes a standard compliant AAS Access Rules to Company A, which Company A simply copies into his AAS Server.
- With respect to security this case is very similar to the “Build My Own” access pattern.

Integration Pattern “Access to External” (IP3)

- Company A accesses an AAS on an external server of Company B.
- Company A accesses AAS from one business partner (bilateral) or accesses AAS from many different business partners (multilateral).
- The external business partner Company B hosts his AAS on his server.
- Company A accesses AAS by the API and uses it in its user applications. If the usage policy from Company B allows it, Company A copies the AAS and store it in its own systems.
- In case of bilateral access, Identity Provider, access token and claims are defined by Company B.
- In case of multilateral access, a common Identity Provider of a dataspace is used. The members of the dataspace have to define access token and claims.
- Access rules need to be defined by Company B to allow access by Company A.

Integration Pattern “Sell But Still Access” (IP4)

- Company A sells a product to Company B.
- Company A has created the AAS and Company B copies the AAS.
- The AAS is hosted on a server of company B.
- Company B defines Identity Provider, access token and claims.
- Company A proposes a AAS Access Rules to Company B, which Company B simply copies into his AAS Server.
- With respect to security this case is similar to the bilateral “Access to External” access pattern.

External and Internal Identity Providers

A Secured System in AAS Security can be Discovery Service, Registry Service, AAS/Submodel Service or AAS/Submodel/ConceptDescription Repository Service.

Figure 10 shows different alternatives of Clients, Identity Providers (IDP) and Secured Systems interacting.

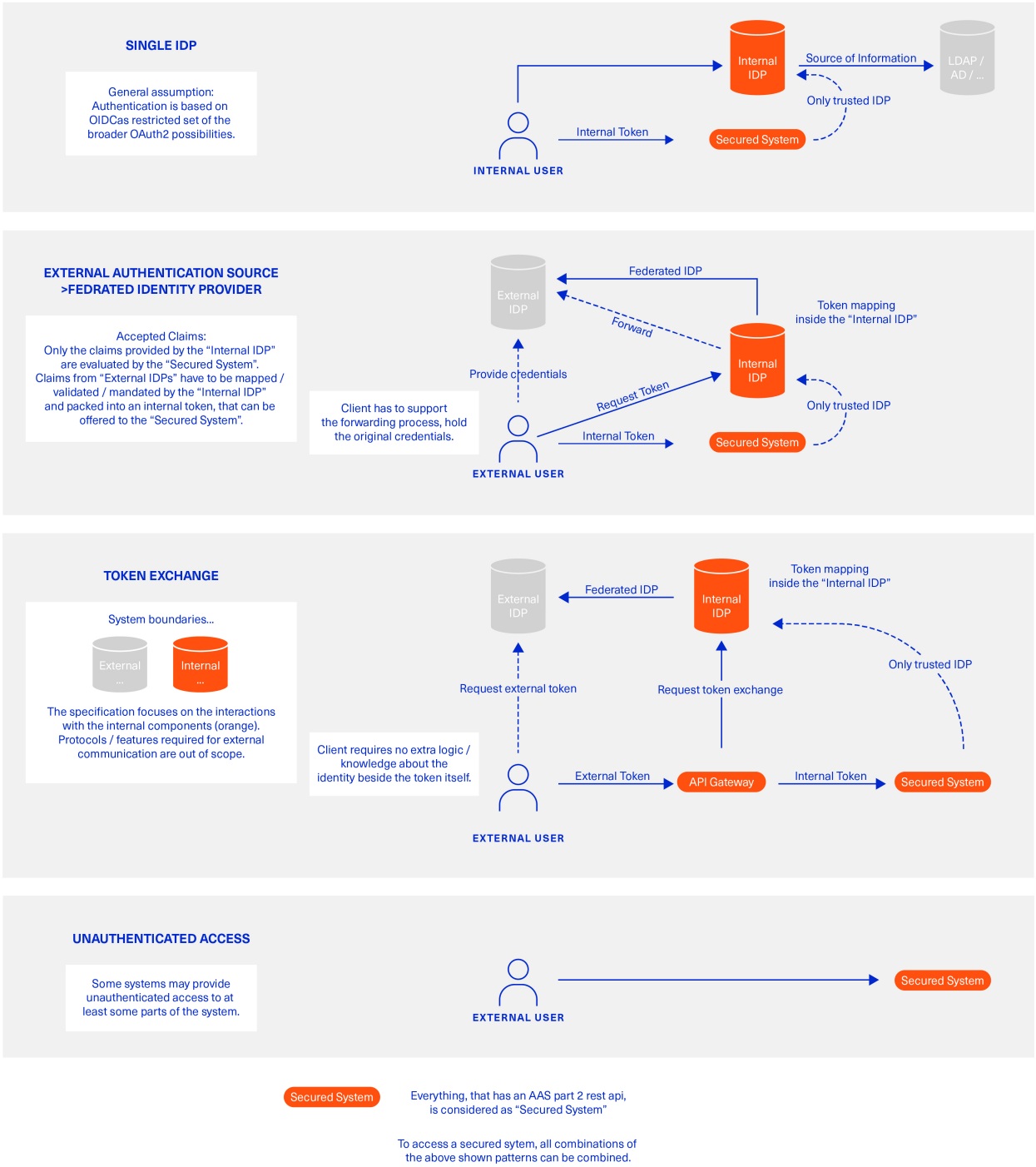


Figure 10. External and internal Identity Providers

Unauthenticated Access

AAS Security supports unauthenticated access to a secured system.
A lot of information is publicly available to human users by web browsers. If such information is provided machine-readable as AAS, it is typically also public without authentication. Examples are the submodels Nameplate or TechnicalData.

Single Identity Provider

AAS Security supports company specific Identity Providers.

In case an internal system is set up, an internal Identity Provider will be used, which can be used by internal user applications.

This might even be used for external user applications, in case a company IT might not allow to support the federated identity providers below. In that case still AAS with AAS Security can be provided, but using a company specific Identity Provider.

Federated Identity Provider

AAS Security supports Identity Providers of dataspaces or of other multilateral company use cases.

In that case, an External User authenticates to an External Identity Provider and receives an external token.

This external token is provided to the Identity Provider of a company and the External User receives an access token. Such federation is possible, when an Identity Provider of a company can check the validity of the external token and trusts the External Identity Provider.

Token Exchange

In case of “Federated Identity Providers”, the external user has to be aware of the “Internal IDP” and has to process (store and forward) the token to the secured system. If the external user requires access to other resources, that are not secured by the “Internal IDP”, he has to deal with multiple access tokens.

To overcome this complexity, RFC 8693 (https://www.rfc-editor.org/rfc/rfc8693) specifies a mechanism, where the external user is able to use its “natural / external” access token to communicate with a secured system under control of the “Internal IDP”.

In cases where a shared AAS/SM Registry references identifiables from different providers, the token exchange protocol enables transparent access to resources of different secured systems under control of different Identity Providers, as long as the Identity Providers trust each other and are able to map the tokens as described in RFC 8693.

From administration point of view, the Federated Identity Provider and the Token Exchange approach enable a system owner to fully control access to their secured systems, because the Identity Provider is managed by the system owner.

Self-sovereign identity wallets provide a way to achieve a distributed identity layer.

Authentication Flows

AAS Security supports any authentication flow which provides signed JSON Web Tokens. Even other formats for Access Tokens might be used.

An authentication flow can be with or without user interaction, depending on the dataspace or use case.

Authentication flows like OAUTH 2.0 or OIDC can be used.

Authentication flows as defined by a specific dataspace or by a specific dataspace protocol can be used.

In any case a signed JSON Web Token is provided by the related Identity Provider.

The following sections describe example flows that can be used.The following terms are used in the description:

Resource owner: A logical entity capable of granting access to a protected resource. When the resource owner is a person, it is referred to as an end-user [RFC 6749]. In the context of AAS, the Resource owner is the AAS Responsible, or a user or organization that has been entitled by the AAS Responsible. In the OAuth 2.0 Client Credential Flow, the client acts on behalf of the resource owner and therefore receives permissions according to the rights of the resource owner.

Resource server: The server hosting the protected resources, capable of accepting and responding to protected resource requests using access tokens. [RFC 6749]

Authorization Server: The server issuing access tokens to the client after successfully authenticating the resource

owner and obtaining authorization. [RFC 6749]

Identity Provider: An entity (usually an organization) that is responsible for establishing, maintaining, securing, and vouching for the identities associated with individuals. [RFC 6973]

Example: OAuth 2.0 Client Credential Grant

The specification of the OAuth 2.0 Framework (RFC 6749) defines Client Credentials flow as:

Client credentials are used as an authorization grant typically when the client is acting on its own behalf (the client is also the resource owner) or is requesting access to protected resources based on an authorization previously arranged with the authorization server.

- The Client Credentials Flow involves an application providing its application credentials.
- The application credentials are client id and client secret.
- This flow is best suited for Machine-to-Machine (M2M) applications, such as CLIs, daemons, or backend services.

The system authenticates and authorizes the application instead of a user.

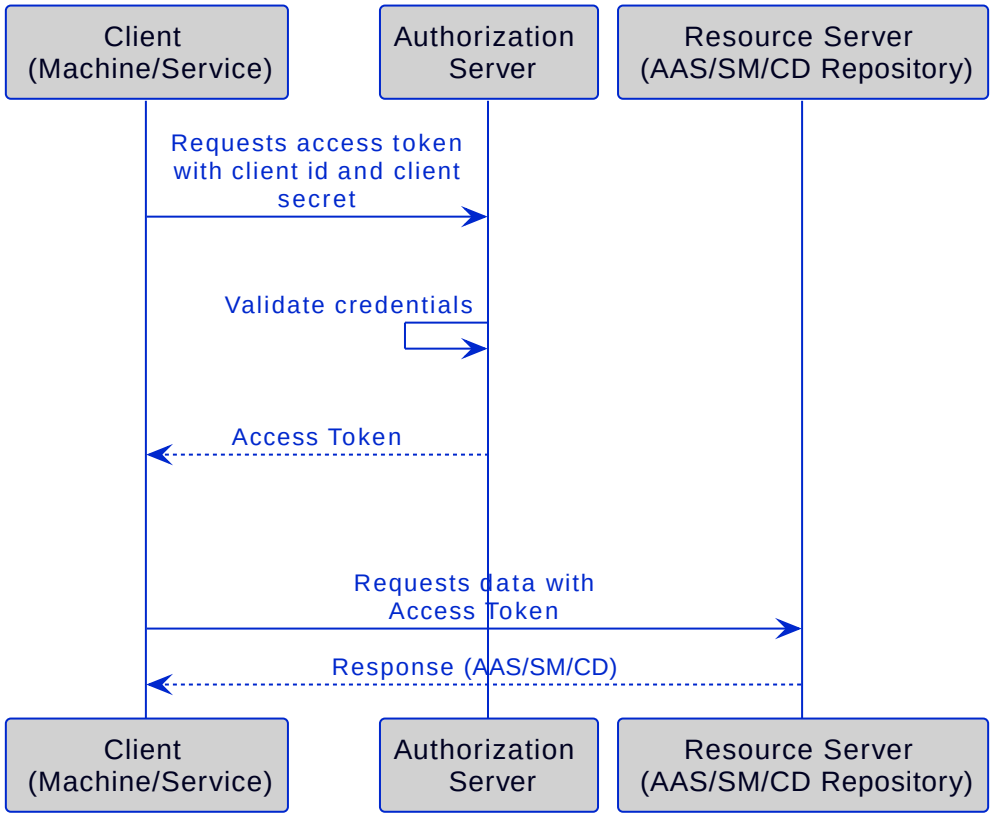


Figure 11. Authentication Flow OAuth 2.0

Figure 11 represents an OAuth 2.0 Client Credentials Grant flow that consists of the following steps:

1. **Client Requests Access Token**:
   - The client (Machine/Service A) sends a request to the Authorization Server (Identity Provider).
   - This request includes the client's credentials, specifically the client ID and client secret.
2. **Authorization Server Validates Credentials**:
   - The Authorization Server receives the request and validates the client's credentials (client ID and client secret).
   - Client secrets are usually long, random strings designed to be difficult to guess. It is commonly an alphanumeric string or Base64 encoded string.

- If the credentials are valid, the Authorization Server generates an access token. If not, an error is returned.

3. **Authorization Server Issues Access Token:**
   - After successful validation, the Authorization Server sends an access token back to the client.
4. **Client Requests Data with Access Token**:
   - The client uses the obtained access token to make a request to the Resource Server (AAS/SM/CD Repository).
   - The access token is included in the request header for authentication and authorization.
5. **Resource Server Responds**:
   - The Resource Server validates the access token.
   - If the token is valid and has the necessary permissions, the Resource Server processes the request and sends the requested data back to the client.
   - The response contains the requested data (AAS/SM/CD).

In the OAuth 2.0 Client Credentials Grant flow the client id and client secret can simply be any string. As per the official specification [RFC 6749] the client id and client secret is any visible (printable) ASCII character (*VSCHAR). The Resource server evaluates and checks the claims in the access token and validates it with the Access Rules, as specified later in this specification.

Example: OpenID Connect

ISO/IEC 26133:2024 defines OpenID connect as:

OpenID Connect 1.0 is a simple identity layer on top of the OAuth 2.0 protocol. It enables Clients to verify the identity of the End-User based on the authentication performed by an Authorization Server, as well as to obtain basic profile information about the End-User in an interoperable and REST-like manner.

Client Credentials Flow in OIDC: While the Client Credentials flow is typically used for authorization (OAuth 2.0), OIDC can also be used in a similar manner but is more often used for scenarios involving (human) user authentication. However, the main distinction comes when OIDC is used in other flows, such as the Authorization Code flow, where it provides ID tokens to assert the identity of the end-user.

ID Token: The ID token is a JSON Web Token (JWT) that contains user profile information (claims) and is signed by the identity provider.

Instead of using user context, the client credential flow uses application context, and an ID token is not issued in this scenario. Only access tokens can be obtained by applications.

The following sequence diagram explains the OpenID Connect in OAuth2 Authorization code grant flow:

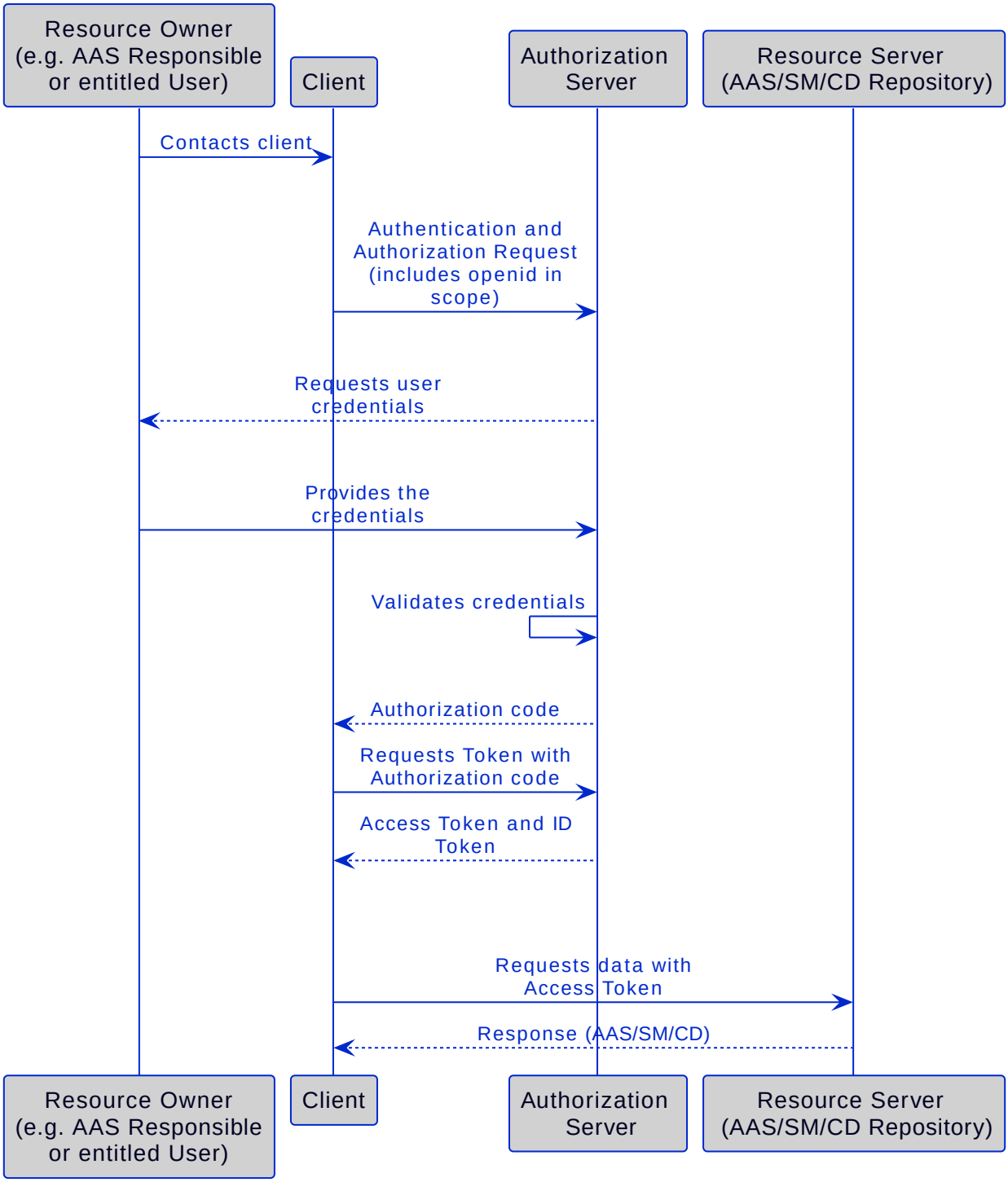


Figure 12. Authentication Flow OpenId Connect

Here's a step-by-step explanation of the flow depicted in Figure 12:

1. **Authentication and Authorization Request:** The client application initiates an authentication and authorization request to the Authorization Server (Identity Provider). This request includes the openid scope, indicating that the client is requesting authentication using OpenID Connect.
2. **User Provides Credentials**: The user is prompted to provide his credentials (e.g., username and password) to the Authorization Server.
3. **Authorization Code**: Upon successful authentication, the Authorization Server issues an authorization code and redirects the user back to the client application with this code.
4. **Request Token with Authorization Code**: The client application sends a request to the Authorization Server to exchange the authorization code for tokens. This request includes the authorization code received in the previous step.

5. **Access Token and ID Token**: The Authorization Server validates the authorization code and, if valid, issues an access token and an ID token to the client application.
6. **Request Data with Access Token**: The client application uses the access token to request data from the Resource Server (AAS/SM/CD Repository). The access token is a credential that allows the client to access protected resources on behalf of the user if authorized.
7. **Response**: The Resource Server validates the access token and, if valid, responds with the requested data (AAS/SM/CD).

Tokens:

- **Authorization Code:** A temporary code issued by the Authorization Server after the user successfully authenticates. It is used to obtain the access token and ID token.
- **Access Token:** A token that allows the authorized client to access protected resources on behalf of the user.
- **ID Token:** A token that contains information about the user (such as user ID, email) and is used to verify the user's identity.

AAS are used both in contexts with and without User, i.e. often in Machine-to-Machine (M2M) applications. For M2M typically only an Access Token and no ID Token is used. In both contexts the Authorization Server creates an Access Token with claims. The AAS server will evaluate and check these claims by the Access Rules as specified later in this specification.

Example: DataSpace Protocol

According to the Dataspace Protocol (DSP) [7], a Dataspace is "A set of technical services that facilitate interoperable Dataset sharing between entities." The specification goes on to define interactions between a Data Provider and Data Consumer to expose metadata, agree on conditions for exchange and execute the transfer of Datasets. Asset Administration Shell resources can be such Datasets.

A Dataspace decides which AAS Service Specifications of AAS Part 2 it wants to use, e.g. using both AssetAdministrationShellRegistryServiceSpecification and SubmodelRepositoryServiceSpecification together with AssetAdministrationShellDescriptors and included SubmodelDescriptors. Specific Dataspaces define conventions on the standardized exposure of these resources for Data Consumers to discover and retrieve.

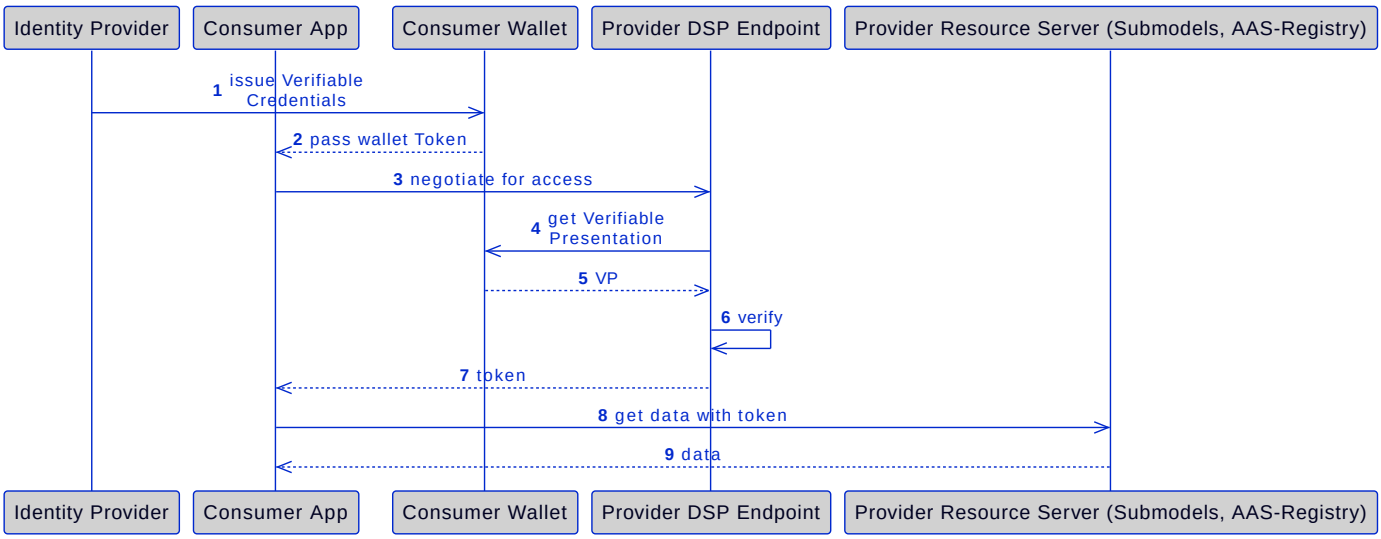


Figure 13. Dataspace Protocol negotiation with integrated SSI-based authentication flow for credential presentation

A Provider will define access conditions encoded as ODRL-Policies that he expects a potential Consumer to comply with. To authorize against these Policies, a Consumer provides access to long-living claims signed by a commonly trusted party (Identity Provider, step 1). These claims will usually follow the Verifiable Credential Data Model embedded in a jwt (see Example Verifiable Credential for authentication and authorization against Dataspace Protocol endpoints):

Example Verifiable Credential for authentication and authorization against Dataspace Protocol endpoints


```
{
  "iss": "did:web:dataspace-identity-issuer.com",
  "vc": {
    "@context": [
      "https://www.w3.org/2018/credentials/v1",
      "https://w3id.org/dataspace-specific/credentials/context.json"
    ],
    "id": "1f36af58-0fc0-4b24-9b1c-e37d59668089",
    "type": [
      "VerifiableCredential",
      "SpecificCredentialType"
    ],
    "issuer": "did:web:dataspace-identity-issuer.com",
    "issuanceDate": "2021-06-16T18:56:59Z",
    "expirationDate": "2022-06-16T18:56:59Z",
    "credentialSubject": {
      "id": "did:web:consumer.com",
      "holderIdentifier": "SomeUniqueIdentifier"
    }
  },
  "jti": "ba1300d8-8a78-40b4-8749-60d84fc3ab97",
  "sub": "did:web:consumer.com",
  "iat": 1737276930,
  "nbf": 1737363330,
  "exp": 1768985730
}
```


A Data Provider will verify those claims by validating the VC's signature using the Identity Provider's public key (step 5). This step ensures the authenticity of the claims. If the claims match the Data Provider's expectation for a particular Policy Constraint, the Data Provider will return information about accessing the exposed AAS resources. This will usually include the URL and a short-living access token as well as a refresh token (step 6). The Tokens' format is usually irrelevant for interoperability as it is issued and validated by the same Participant and remains opaque to the Consumer. Only scenarios where the Policy Decision Point and Policy Enforcement Point are implemented in separate runtimes, possibly from different vendors, require standardization of such tokens. Access concludes by presenting said short-lived token to the server hosting the AAS-resource (step 7).

To strike a balance between granularity and public meta-data exposure, AAS resources are exposed in a granular manner masking implicit information like the number of Submodels or AAS-Descriptors. The AAS-Registry API completes the Data Consumer's discovery-sequence by linking the Submodels from distributed data sources while at the same time protecting the AAS-Descriptor objects from the public.

The AAS data server and the AAS registry will evaluate and check the claims in the token by the Access Rules as specified later in this specification.

IEC 63278

Figure 14 shows a detailed overview of AAS as defined in IEC 62378-1. The yellow comments have been added and describe the scope of this document in relation to the defined entities in the standard. Signatures are defined in the REST API specification.

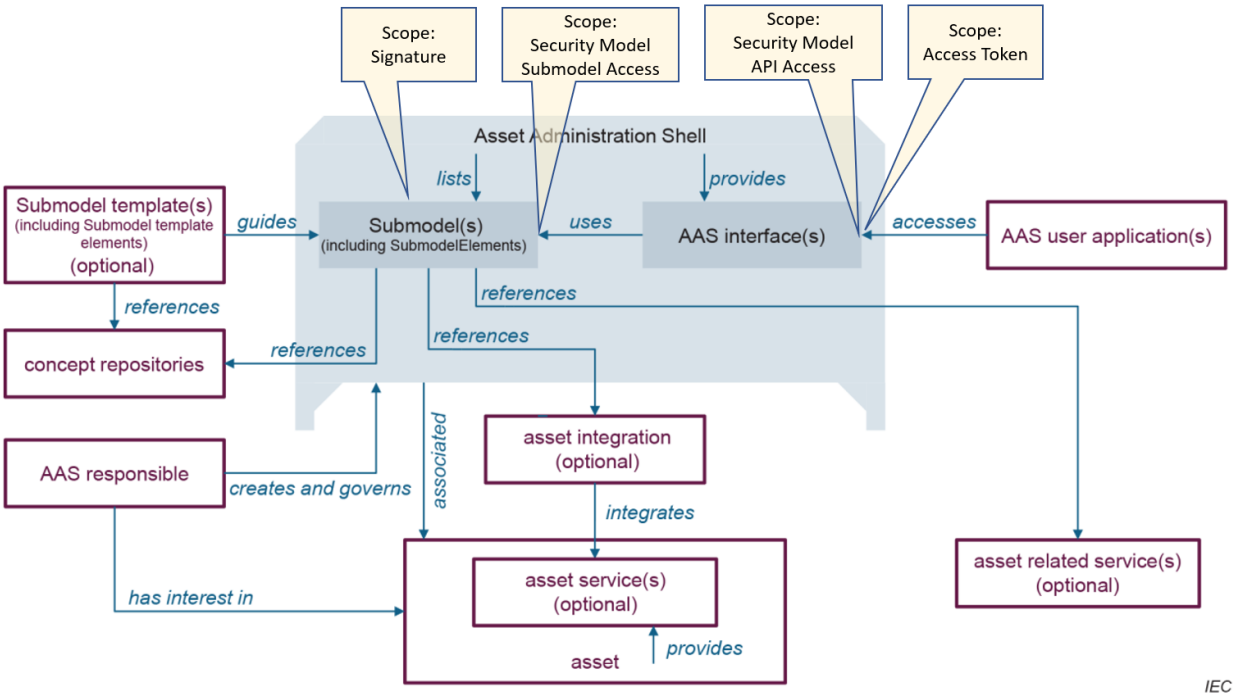


Figure 14. Detailed overview of Asset Administration Shell and related entities. Based on Figure 4 of IEC 63278-1 (Reproduced by permission of DKE, German Member of IEC (www.dke.de). For the valid edition see the latest publication of IEC or DIN EN IEC and www.dke.de and www.vde-verlag.de)

Security objectives

Overview

Information security has traditionally focused on achieving three objectives, confidentiality, integrity, and availability, which are often abbreviated by the acronym CIA. IEC TS 62443-1-1:2009 observes that these priorities are sometimes completely inverted, e.g., availability, integrity, confidentiality (AIC) in case of some industrial automation scenarios. In AAS context, AAS is used in a variety of different use cases. While some domains suggest a specific ordering or priority of security objectives, the order of the following subsections on integrity, confidentiality, and availability is deliberate. The AAS responsible SHOULD consider the three objectives without predefining their priority and use a risk-based approach to determine and balance the priority of the individual security objectives for their intended use cases and applications.

Integrity

Integrity of AAS interface behaviour: AAS SHALL ensure the logical correctness and reliability of data represented by the AAS as well as structure and occurrence of the data represented by the AAS. AAS SHALL ensure that data is provided in a consistent and reproducible way. The AAS interface SHALL ensure the authenticity, reliability, and security of API interactions, preventing unauthorized access and abuse. Execution of operations and access to data SHALL behave in a predictable manner.

> Note: The amount of data represented by the AAS being visible and the AAS services being available are subject to access control (Authorization enforcement, [access-rule-model:::access-rule-model]). That means, the user experience depends on the privilege level of the AAS user application user. For example, it is possible that specific Submodels are only visible for privileged repair personnel. Using ABAC, the amount of information can also depend on additional criteria, like the operation status of a manufacturing system.

Integrity and accountability of data: The AAS user application user expects the data of the AAS provided via the AAS interface to be correct and reliable. Protection of integrity and accountability of AAS data SHALL consider the following:

- Protection against the entry of data by an unauthorized AAS user application user. Entry/modification of data SHALL only be possible for an authorized user.
- It SHOULD be possible to detect unintentional change of data represented by the AAS and allow to trace the originator of data.
- Protection against manipulation of data represented by the AAS. Data represented by the AAS SHOULD be protected against unauthorized manipulation including the manipulation of origin/authorship. This is especially important if data represented by the AAS is handled (e.g., storage, retrieval, aggregation) by 3rd parties who are not the original source of that data.
- Protection against the manipulation of the AAS structure. It is possible that moving Properties or SubmodelElements between different SubmodelElementCollections results in incorrect interpretation of the provided data.
- Protection against the manipulation of AAS management information and auxiliary information used by systems realizing an AAS. Auxiliary information includes all information relevant for the correctness of data represented by the AAS and the correct functionality of the AAS interface. This especially includes the information relevant for AAS access control.

Integrity of relation between data and asset: The relation between an asset and the data stored in the AAS SHALL be protected. It is important to know that AAS data belongs to the asset as identified by the asset identifier recorded in the AAS management information.

Integrity of relation between services and asset: There SHALL be no incorrect relations between an asset and asset related services provided by the AAS. It is essential that services referenced by AAS Submodels affect the intended asset and that the behaviour of the service is described correctly.

Confidentiality

Data exposure and data leakage protection: AAS SHALL provide assurance that data represented by the AAS is not disclosed to unauthorized entities. Access to information considered sensitive by the AAS responsible SHALL be controllable by well-defined rules determining the visibility of data represented by the AAS. Access to sensitive information represented by the AAS SHALL only be granted after authentication of the requestor (Identification and authentication of AAS user application users, Identification and authentication of AAS user applications and AAS interface). This applies to both,

- actual values,

as well as the

- structure and meta-data represented by the AAS (topology described Submodels, SubmodelElements, SubmodelElementCollections, …).

Availability

Availability of asset data: The AAS interface SHALL be able to process requests in a reasonable time interval. The AAS responsible SHALL define the quality of service provided.

Data represented by the AAS returned via the AAS interface SHALL be current. It SHALL NOT be possible to maliciously hide data stored in an AAS. It SHALL be possible to determine whether data is sufficiently recent (lost updates) or data has been intentionally removed (malicious rollbacks).

> Note: This requirement does not affect the situation where data represented by the AAS is intentionally configured to be visible or invisible for AAS user application users depending on their privilege level (Authorization enforcement, [access-rule-model:::access-rule-model]).

AAS Security requirements

General

AAS security describes a holistic approach to address security concerns related to AAS. The requirements for AAS security have been derived from the foundational requirements given for systems and components in IEC 62443-3-3 and IEC 62443-4-2 respectively. The requirements mainly address the protection of the structure and data represented by the AAS (IDTA 01001 [1]) and made available to the AAS user application via the AAS interface (IDTA 01002 [2]). This document also describes the role of the AAS responsible for security governance. The selection of requirements from IEC 62443 applicable for AAS security follows the analysis in [annex::requirements_mapping:::_assessment_of_iec_62443_requirements_for_aas_structure] provided as annex.

Security requirements concerning the AAS structure and interface

Overview

Following subclauses are based on the seven foundational requirements defined in the IEC 62443 series.

- FR1: Identification and authentication control.
- FR2: Use control.
- FR3: System integrity.
- FR4: Data confidentiality.
- FR5: Restricted data flow.
- FR6: Timely response to events.
- FR7: Resource availability.

Requirements regarding Authentication Control (FR1)

Identification and authentication of AAS user application users

Before granting access to protected data represented by the AAS or restricted services, the AAS interface SHALL ensure the identification and authentication of the AAS user application user. Authentication of the AAS user application user SHALL be based on the authentication of information provided by the AAS user application on behalf of a user. The subject attributes conveyed to authenticate the AAS user application user SHALL uniquely determine the subject credentials used for subject authentication. The identity vetting process used to issue subject credentials SHOULD be sufficient to hold the identified AAS user application user accountable (see Integrity, Public key infrastructure).

If the AAS user application user is a NPE the AAS user application MAY be considered as the user. This especially accounts for AAS user applications with unattended operation. If there is no distinction between AAS user application user and AAS user application the identification and authentication of the AAS user application user MAY be replaced by the identification and authentication of the AAS user application (Identification and authentication of AAS user applications and AAS interface).

Additional information provided by the AAS user application can be used to provide extended information on the usage context. All information provided by the AAS user application can be used for authorization decisions.

Attributes describes an approach to provide information for authorization via ABAC attributes in an authenticated way.

> Note: Derived from IEC 62443-3-3 SR 1.1, IEC 62443-4-2 CR 1.1.

Identification and authentication of AAS user applications and AAS interface

Identification and authentication of AAS user applications and AAS interface provides a basis for integrity protection

(Integrity) and confidentiality protection (Confidentiality) for data represented by the AAS exchanged via the AAS interface.

- Before granting access to protected data represented by the AAS or restricted services, the AAS interface SHALL ensure the identification and authentication of the AAS user application. Authentication of the AAS user application SHALL include the authentication of ABAC attributes provided by the AAS user application. The content of the ABAC attributes can affect the behaviour of the AAS interface. All ABAC attributes provided by the AAS user application can be used for authorization decisions.
- If requested by the AAS user application the AAS interface SHOULD be able to prove its identity towards the AAS user application.

Identification and authentication of the AAS user application and the AAS interface can be achieved by different means, depending on the underlying communication technology and required security level (Public key infrastructure).

> Note: In case of TCP/IP based communication (mutual) TLS can be used to provide secure identification between the communication partners.

If an AAS user application is under the complete control of the AAS user application user, the identification and authentication of the AAS user application MAY be replaced by the identification and authentication of the AAS user application user (Identification and authentication of AAS user application users). This especially accounts if a commodity application (e.g., web browser) which functions as AAS user application.

> Note: Derived from IEC 62443-3-3 SR 1.2, IEC 62443-4-2 CR 1.2.

Account management

The AAS interface SHALL be able to use accounts for AAS user applications and AAS user application users from an existing service or provide a built-in account database.

If a built-in account database AAS implementation SHALL provide means to properly protect account information and enforce authorization for the modification of account information. If a detached service is used to authenticate the user, the AAS responsible SHALL ensure that the account information is aligned between the AAS and the detached service (e.g., user roles assigned by a dedicated service need to match the roles evaluated by the AAS interface).

Account management MAY be made accessible via the AAS interface.

> Note:¬Derived from IEC 62443-3-3 SR 1.3, IEC 62443-4-2 CR 1.3.

Identifier management

All entities within the AAS or interacting with the AAS SHALL be well defined and have unambiguous identifiers. The definition of an identifier includes its syntax/format and namespace management. The AAS responsible SHALL define a policy for identifiers used for:

- AAS instances and AAS interfaces
- Semantic identifiers and concept repositories
- Assets
- AAS objects (Objects)
- AAS user applications
- AAS user application users

Note: IEC 61406-1:2022 is commonly applied for the identification of physical assets.

Note: Derived from IEC 62443-3-3 SR 1.4, IEC 62443-4-2 CR 1.4.

Public key infrastructure

Authentication of AAS user applications and AAS user application users SHALL support and facilitate the use of secure identities for authentication (Identification and authentication of AAS user application users, Identification and authentication of AAS user applications and AAS interface) and provide the capability to integrate into a system used to manage these secure identities across security domains of the different stakeholders interacting along a value chain.

Examples for secure identity infrastructures are:

- PKI X.509 identities (ITU-T Recommendation X.509 (2005) [10]).
- Decentralized Identifiers (DID) and Verifiable Credentials (VC) (Decentralized Identifiers (DIDs) v1.0 [11]).

Derived from IEC 62443-3-3 SR 1.8, IEC 62443-4-2 CR 1.8.

Authenticator feedback

In case a request sent from the AAS user application to the AAS interface cannot be processed, AAS interface SHALL NOT disclose any information that would allow conclusions to be drawn about the content or structure of the AAS nor conclusions on why a specific request was rejected.

Additional information to support trouble shooting or debugging MAY only be made available after additional authorization for this function.

Note: IEC 62443-4-2 CR 1.10.

Requirements regarding Use Control (FR2)

Authorization enforcement

Authorization enforcement happens at the AAS interface. The authorization of AAS user application users accessing the AAS interface SHALL be based on a defined set of rules. Authorization rules MAY be based on ABAC attributes provided by the AAS user application during authentication.

Details on how this requirement is to be met by the information model are outlined in [access-rule-model:::access-rule-model].

Note: Derived from IEC 62443-3-3 SR 2.1, IEC 62443-4-2 CR 2.1.

Auditable Events

The AAS interface SHALL create audit events. As a minimum the events listed in Table Table 1 SHALL be kept in an AAS interface audit trail.

Table 1. Security Events

	Event	Description
S1	SecurityUnauthorizedAccessEvent	Any attempt of unauthorized access or modifiy the AAS (Authorization enforcement). The event SHALL contain information on the origin of the unauthorized access (e.g., network source, subject name) and on the target AAS object (Objects).
S2	SecurityIdentityConfigChange	Modification of identity related configuration of the AAS (Identifier management, Public key infrastructure), e.g., • own credentials used by the AAS; • 3rd parties trusted by the AAS.
S3	SecurityAccountConfigChange	Modification related to AAS user application or AAS user application user accounts (Account management).
S4	SecurityAccessRuleConfigChange	Modification of access control rules.

If accountability is a concern, the events listed in Table Table 2 SHALL be reported. Each accountability related event SHALL provide:

- Which AAS object (Objects) is affected
- Whether the operation could be completed successful

Table 2. Accountability Events

	Event	Description
A1	AASInformationCreate	Addition of new information (AAS, Submodel, SubmodelElement)
A2	AASInformationUpdate	Update of information of an existing AAS object (Objects).
A3	AASInformationDelete	Removal of information (AAS, Submodel, SubmodelElement)

Note: Derived from IEC 62443-3-3 SR 2.8, IEC 62443-4-2 CR 2.8.

Timestamps

Each auditable event SHALL contain a time stamp of the event with precision at least to seconds.

Note: Derived from IEC 62443-3-3 SR 2.11, IEC 62443-4-2 CR 2.11.

Non-repudiation

Each auditable event shall contain information to determine which AAS user application took a particular action on the AAS. Providing detailed information on a particular event (e.g., actual user identity) may be delegated to the AAS user application. Depending on the intended use case the AAS responsible shall ensure that the available information is sufficient to resolve any emerging dispute.

Note:¬Derived from IEC 62443-4-2 CR 2.12., IEC 62443-3-3 SR 2.12.

Requirements regarding AAS Integrity (FR3)

Information integrity

The integrity of the structure of the AAS itself and information represented by the AAS, for example, Submodels, SubmodelElements SHALL be protected.

This requirement is relevant for two aspects of the AAS security model:

- Data at rest protection, i.e., storage of information represented by an AAS SHALL be protected against tampering.
- Data in transit protection, i.e., ensure authenticity of source and recipient (Integrity, Identification and authentication of AAS user applications and AAS interface) and protect integrity of information delivered via the AAS interface while in transit to the AAS user application.

Note: In case of TCP/IP based communication TLS can be used to ensure information integrity for data in transit.

Note: Derived from IEC 62443-3-3 SR 3.1, IEC 62443-4-2 CR 3.1 and IEC 62443-3-3 SR 3.4, IEC 62443-4-2 CR 3.4.

Input validation

The AAS metamodel, Submodel templates, and concept repositories provide information supporting input validation checks for asset related information. The AAS interface SHOULD implement checks to ensure that data handed into or retrieved from the Asset Administration Shell for either storage or further processing meets those constraints.

From a security perspective, the AAS Responsible SHOULD discourage the deployment of Submodels which are not guided by a Submodel template.

Strict use of Submodel templates for Submodels and concept repositories for Submodel template elements respectively SubmodelElements can support the validation of the AAS structure and content.

Note: Derived from IEC 62443-3-3 SR 3.5, IEC 62443-4-2 CR 3.5.

Error handling

An AAS constitutes a large aggregation of data and even the acquisition of meta-information is considered a potential goal of attackers. A potential attack is sending brute-force trial-and-error requests and using the error codes returned

as oracle for business relevant information. Examples are information on whether a specific asset exists, amount of assets, existence of a specific Submodel.

In case a request sent from the AAS user application to the AAS interface cannot be processed, AAS interface SHOULD NOT disclose any information that could be exploited by adversaries to attack the AAS. Especially, the AAS interface SHOULD NOT disclose any information that would allow conclusions to be drawn about the content or structure of the AAS.

Note: Derived from IEC 62443-3-3 SR 3.8, IEC 62443-4-2 CR 3.8.

Session integrity

The AAS interface SHALL only accept ABAC attributes from the AAS user application (Identification and authentication of AAS user application users) which are properly bound to the respective communication context/channel (Identification and authentication of AAS user applications and AAS interface, Information integrity) and the information returned needs to be bound to that same communication context/channel (channel binding).

Note: Derived from IEC 62443-4-2 CR 3.8, IEC 62443-3-3 SR 3.8

Asset roots of trust management

AAS Submodels can be used to distribute roots of trust to facilitate the authentication and communication with respect to the managed assets. For instance, an AAS Submodel can contain current root certificates of product suppliers or asset owners to support authentication of assets equipped with cryptographic credentials like manufacturer device certificates or IEEE 802.1AR device identifiers.

Note: Derived from IEC 62443-4-2 CR 3.12.

AAS roots of trust management

AAS Responsible SHALL define roots of trust for:

- Protection and authentication of communication (Identification and authentication of AAS user applications and AAS interface, Information integrity).
- Authentication and authorization services used for the realization of an AAS (Identification and authentication of AAS user application users, Identification and authentication of AAS user applications and AAS interface, Attributes).

Note: Derived from IEC 62443-4-2 CR 3.13.

Requirements regarding Data Confidentiality (FR4)

Information confidentiality

The confidentiality of information represented by an AAS SHALL be protected. This information includes both meta-information, e.g., which Submodels are available, and content provided via AAS, for example, Submodels, SubmodelElements.

This requirement is relevant for two aspects of the AAS security model:

- Data at rest protection, i.e., storage of information represented by an AAS SHALL provide protection against unintended data disclosure.
- Data in transit protection, i.e., ensure authenticity of source and recipient (Integrity, Identification and authentication of AAS user applications and AAS interface) to avoid exposure of data to unintended recipients and protect confidentiality of information delivered via the AAS interface while in transit to the AAS user application (Confidentiality).

Note: In case of TCP/IP based communication TLS can be used to ensure confidentiality protection for data in transit.

Confidentiality for data at rest can be implemented by different means, depending on the required security level.

Measures include:

- Implementation of access control (Authorization enforcement).
- Use of protected storage systems and data bases relying on cryptographic encryption mechanisms (symmetric and asymmetric) on the actual data.
- Providing information by-reference instead of by-value. The responsibility to protect the information is delegated to the referenced data source. Delegation is an option to be considered in case AAS is not able to meet the required protection level.

Note: Derived from IEC 62443-3-3 SR 4.1, IEC 62443-4-2 CR 4.1

Use of cryptography

If no details are mentioned on required cryptographic algorithms or strength used to implement AAS security functionality, the latest version of [ENISA Report] or [NIST SP 800-78-5] SHOULD be used as a reference.

Note: Derived from IEC 62443-3-3 SR 4.3, IEC 62443-4-2 CR 4.3

Requirements regarding restricted data flow (FR5)

Application partitioning

The definition of AAS structures SHOULD consider separation of content based on associated risk, relevant security objectives, or other criteria, operational function, required access (for example, least privilege principles) or responsible organization.

Restricted data flow is an important concept of IEC 62443.

Structuring of AAS information is described in IDTA 01001 [1]. AAS, Submodels, and SubmodelElements SHOULD be structured in a way that reflects different security levels applying to individual information offered by the respective substructure, e.g., by confining sensitive information into separate Submodels or SubmodelElements.

Security attributes such as access control rules (Access Rule Model) SHALL be assignable at the level of the entire AAS, individual Submodels, and SubmodelElements.

The elements of the AAS can be distributed across different entities (e.g., local device, edge device, on-premise cloud service, public cloud) according to their sensitivity. Each entity providing a different level of protection according to its security capabilities.

Example: While Submodels containing insensitive information are hosted directly on an embedded device with constrained security capabilities the access to Submodels containing more sensitive information or asset related services are referred (via References) to a backend service or edge device supporting more sophisticated security mechanisms.

Access to the AAS interface can be restricted to defined AAS user applications.

Note: Derived from IEC 62443-3-3 SR 5.4.

Requirements regarding Timely Response to Events (FR6)

Audit log accessibility

Auditable events (Authorization enforcement) can be made available via the AAS interface, e.g., as a dedicated Submodel.

Continuous monitoring

Auditable events (Auditable Events) SHALL be provided in a machine processable format.

Requirements regarding Resource Availability (FR7)

Denial-of-service protection

In order to provide the capability to operate in a degraded state in case of a denial-of-service attack, the AAS interface:

- SHALL reject any AAS user application that is not able to meet the authentication requirement (Identification and authentication of AAS user applications and AAS interface).
- SHOULD reject unauthorized requests as early as possible (cf. Identification and authentication of AAS user applications and AAS interface, Authorization enforcement), for example, by applying upfront access restriction rules which do not depend on AAS authorization mechanisms ([access-rule-model:::access-rule-model]). Potential measures are network protection and restricted data flow (Requirements regarding restricted data flow (FR5)), AAS user application quotas (e.g., limit the amount of requests/data retrieved from a specific AAS user application), proof-of-work provided by the AAS user application (e.g., CAPTCHAs, cryptographic puzzles). These measures can also be used to prevent against data scraping by authorized users.

The AAS can provide information about the actuality (e.g., last update received) of the information to allow AAS user applications.

Note: Versioning of AAS information is described in IDTA 01001 [1]. Versioning information can be used by the AAS user application to decide whether the data is still reliable and to detect data inconsistencies which MAY have occurred due to an attack (e.g., lost updates).

Note: Derived from IEC 62443-3-3 SR 7.1, IEC 62443-4-2 CR 7.1.

AAS recovery and reconstitution

The AAS can support the AAS user application to detect whether information provided is current. The AAS user

application can use this information to determine whether recent updates have been lost due to denial-of-service (DoS) or AAS recovery or reconstitution after an attack.

Example: One example for realization is to provide appropriate timestamp or versioning information on the individual AAS objects (6.2.4.5).

Note: Versioning of AAS information is described in IDTA 01001 [1].

Note: Derived from IEC 62443-3-3 SR 7.4, IEC 62443-4-2 CR 7.4.

Least functionality

The AAS interface SHALL provide the capability to restrict the discovery, exploration, and use of the AAS itself and Submodels contained therein.

The discovery, exploration, and use of Submodels is addressed by Authorization enforcement and [access-rule-model:::access-rule-model].

Derived from IEC 62443-3-3 SR 7.7, IEC 62443-4-2 CR 7.7.

Access Rule Model (normative)

General

The introduction in Chapter *Introduction* has explained in detail the background for AAS Security. The use of Identity Providers and Authentication Flows have been shown, which provide an access token to the client user application. Such access token contains claims as subject attributes which can be used in the Access Rule Model as defined below. In addition, the Access Rule Model contains further ABAC attributes like global attributes (also called environmental attributes) or object attributes.

ABAC attributes assessed by the AAS interface shall originate from a trustworthy source.

Attributes derived from data represented by the AAS (e.g., value of a SubmodelElement) are assumed to be implicitly trusted by the AAS interface.

Global attributes shall originate from a trustworthy source and their authenticity shall be protected (e.g., using a trusted local clock, relying on a trusted time service).

Subject attributes need to be mapped to the subject in an authenticated way. This functionality must be provided by the PIP. The AAS interface relies on access tokens to convey subject attributes Access Tokens.

Access Rule Model gives an overview of the Access Rule Model. Section BNF grammar of Access Rules defines the text serialization of the Access Rule Model. Section JSON Serialization of Access Rule Model defines the JSON schema of the Access Rule Model.

The Access Rule Model allows to define Access Rules in a modular way, so that parts can be reused (see Reusable Definitions).

Whether an Access Rule becomes active or not is decided by Formulas Conditions and Formulas. Only if the overall result of the Formula is valid and true, the Access Rule becomes active. A formula can include nested boolean expressions, comparisons and string operations. A dedicated match operation allows to work with tuples in lists, e.g. specificAssetId.

Access SHALL only be granted if there is (at least) one access rule which allows the requested action on the designated AAS object (Route/Identifiable/Referable/Fragment/Descriptor; see AAS Objects). If there is no access rule, the default behavior shall be to deny access.

This grammar applies to both, respositories and registries. The symbol FieldIdentifier allows to use attributes of AAS, Submodel, SubmodelElement, ConceptDescription, AAS-Descriptor and Submodel-Descriptor. Depending on the type of repository or registry such attributes are allowed or not.

The AAS Query Language and the AAS Access Rules share the same BNF grammar for formula expressions. In addition to the text below, the further details of the formula expressions are explained in IDTA-01002, Query Language .

AAS implementations MAY decide, to which extent the access rule model is used. The access rule model can be used as a guideline which shows, what a client application will expect.

If an AAS implementation allows the exchange of access rules, it SHOULD follow the specification of the access rule model.

AAS implementations MAY decide to support only a subset of the access rules. If an AAS implementation, e.g., only provides a READ profile, only such access rules for defining READ access are required. Access control MAY also be limited to specific subsystems, e.g., AAS repository, submodel repository, concept description repository, AAS registry or submodel registry.

AAS systems are supposed to deal with a large amount of AAS and AAS Submodels. To reach the needed performance and meet memory and storage constraints an AAS implementation MAY translated the access rules into a representation that meets the needs of the respective implementation as long as the behavior complies to the stated rules.

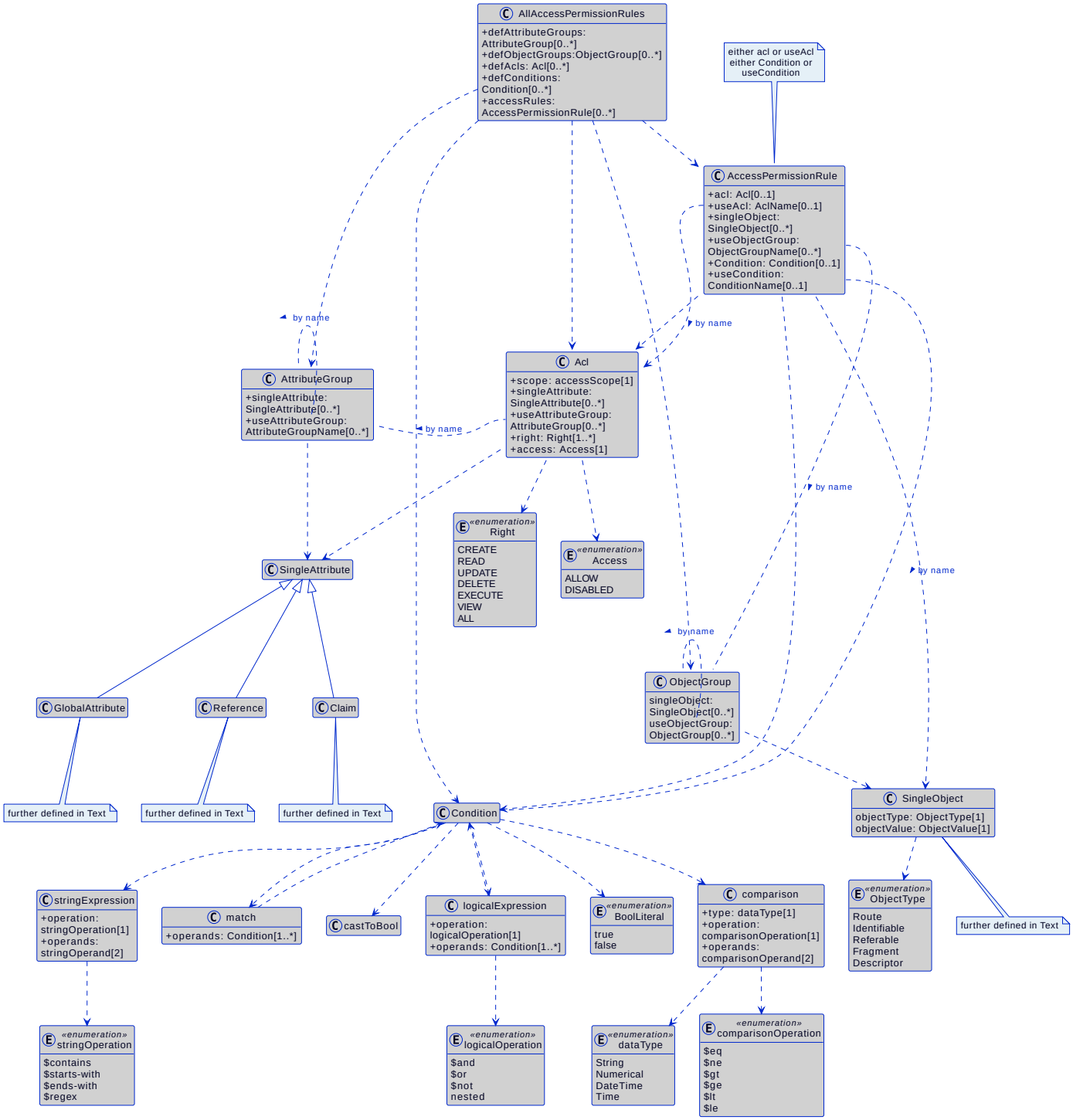


Figure 15. Access Rule Model

Access Tokens

The evaluation of access rules to determine access to the AAS relies on access tokens to convey subject attributes. An access token is an authenticated set of claims about ABAC attributes that were mapped to the subject accessing the AAS. The access token is typically obtained using an Identity Provider. The access token is presented to the AAS along with the API request.

An AAS Data Server trusts one or several Identity Providers. An Identity Provider MAY also give the possibility to federate identities to/from other Identity Providers.

Trust means, that for such Identity Provider(s) an AAS Data Server can verify the public key of the Identity Provider. Such public key MAY be available directly as a key, by a public certificate or by a verifiable credential.

The policy of a dataspace SHALL define which access tokens are trusted by the AAS.

The policy of a dataspaces SHALL define the set of subject ABAC attributes that are used as claims to determine authorization.

Access tokens used for operations resulting in modification of information or disclosure of non-public information provided by the AAS shall use a state-of-the-art, cryptographically strong asymmetric signature scheme to authenticate the access token Use of Cryptography.

AAS Security uses signed JWT (JSON Web Token) bearer tokens as Access Tokens, as they are defined in RFC 7519.

The JWT is digitally signed according to RFC 7519 with the private key of the Identity Provider. The public key used as trust anchor needs to be accessible by the AAS server, such that the JWT can be verified.

Other properties like the lifetime of the token or Proof-of-Possession (RFC 7800) properties SHOULD be determined based on the use case.

AAS Security only requires the standard claims of the RFC 9068 Chapter 2.2 Data Structure for the JWT, i.e “iss” (Issuer), “exp” (Expiration time), “aud” (Audience), “sub” (Subject), “client_id” (Client identifier), “iat” (Issued at), “jti” (JWT ID).

The specific requirements on “iss”, “sub” and “aud” SHOULD be defined by the dataspace policy.

All claims in an Access Token can be used in ABAC access rules as attributes and as described in the Access Rule Model below.

Access rules can also be defined for anonymous access, i.e. no Access Token exists.
This is especially important for the Digital Product Passport to be able to scan the QRCODE according to IEC 61406 with a normal mobile phone camera.

Further definitions can be made by dataspaces. A specific dataspace MAY define the technology how to get an access token and MAY also define the attributes which can be used in the ABAC access rules. Examples are discussed in Identity Providers, Authentication Flows.

This both assures interoperability and makes it possible to integrate new dataspaces in the future.

Grammar for Access Rule Model

BNF grammar of Access Rules

The following BNF (Backus-Naur-Form) grammar defines the AAS Access Rule Model in a technology neutral form. The grammar includes whitespaces, so that the examples in Examples of Access Rules in text serialization can be validated.

The Security Model is explained step by step in Explanation of the Access Rule Model.

Examples can be found in Examples of Access Rules in text serialization.

Further informative examples for RIGHTS, ROUTEs and HTTP requests can be found in Examples of ROUTEs and HTTP requests.

The BNF grammar artifacts are tested in the GitHub Actions CI/CD workflow "Validate spec artifacts". This workflow runs the repository validation script tools/validate_spec_artifacts.py. The script checks all BNF files for balanced delimiters, valid rule definitions and references to defined rules. It also checks the BNF examples in the specification for balanced delimiters and validates model identifiers used in these examples against the related JSON Schema definitions.

BNF Playground can be used for manual testing of the BNF grammar and examples.

The tested BNF is:

```
<AllAccessPermissionRules> ::=
    <AllAccessPermissionDefs>
    ( <AccessPermissionRule> <Ws> )*

<AllAccessPermissionDefs> ::=
    <AllAccessDefAttributes>
    <AllAccessDefAcls>
    <AllAccessDefObjects>
    <AllAccessDefFormulas>

<AllAccessDefAttributes> ::=
    ( "DEFATTRIBUTES" <Ws> <StringLiteral> <Ws> <AttributeGroup> <Ws> )*

<AllAccessDefAcls> ::=
    ( "DEFACLS" <Ws> <StringLiteral> <Ws> <ACL> <Ws> )*

<AllAccessDefObjects> ::=
    ( "DEFOBJECTS" <Ws> <StringLiteral> <Ws> <ObjectGroup> <Ws> )*

<AllAccessDefFormulas> ::=
    ( "DEFFORMULAS" <Ws> <StringLiteral> <Ws> <LogicalExpression> <Ws> )*

<AccessPermissionRule> ::=
    <AccessAccessRule>
    <AccessObjects>
    <AccessFormula>
    <Filter>

<AccessAccessRule> ::=
    "ACCESSRULE:" <Ws> ( <ACL> | <UseACL> ) <Ws>

<AccessObjects> ::=
    "OBJECTS:" <Ws>
    ( <SingleObject> <Ws> )*
    ( <UseObjectGroup> <Ws> )*

<AccessFormula> ::=
    ( "FORMULA:" <Ws> <LogicalExpression> <Ws> ) |
    ( <UseFormula> <Ws> )

<Filter> ::=
    (
        ( "FILTER:" <Ws> <SecurityQueryFilter> <Ws> )?
        ( "FILTERLIST:" <Ws> ( <SecurityQueryFilter> <Ws> )* )?
    )

<SecurityQueryFilter> ::=
    ( "FRAGMENT:" <Ws> <FieldIdentifierFragment> <Ws> )
    (
        ( "CONDITION:" <Ws> <LogicalExpression> <Ws> ) |
        ( <UseFormula> <Ws> )
```

```
    )

<ACL> ::=
    <AclAttributes>
    "RIGHTS:" <Ws> <Right> <Ws> ( <Right> <Ws> )*
    "ACCESS:" <Ws> <Access> <Ws>

<AclAttributes> ::=
    "ATTRIBUTES:" <Ws>
    ( <SingleAttribute> <Ws> )*
    ( <UseAttributeGroup> <Ws> )*

<UseACL> ::= "USEACL" <Ws> <StringLiteral> <Ws>

<Right> ::= "CREATE" | "READ" | "UPDATE" | "DELETE" | "EXECUTE" | "VIEW" | "ALL"

<Access> ::= "ALLOW" | "DISABLED"

<SingleAttribute> ::= <ClaimAttribute> | <GlobalAttribute> | <ReferenceAttribute>

<ClaimAttribute> ::= "CLAIM" <Ws> "(" <Ws> <ClaimLiteral> <Ws> ")"

<GlobalAttribute> ::=
    "GLOBAL" <Ws> "(" <Ws>
    ( "LOCALNOW" | "UTCNOW" | "CLIENTNOW" | "ANONYMOUS" )
    <Ws> ")"

<DateTimeGlobalAttribute> ::=
    "GLOBAL" <Ws> "(" <Ws>
    ( "LOCALNOW" | "UTCNOW" | "CLIENTNOW" )
    <Ws> ")"

<ReferenceAttribute> ::=
    "REFERENCE("
        "$aas" <IdentifierInstance> "#" <FieldsAAS> |
        "$sm" <IdentifierInstance> "#" <FieldsSM> |
        "$cd" <IdentifierInstance> "#" <FieldsCD> |
        "$sme" <IdentifierInstance> "." <IdShortPath> "#" <FieldsSME>
    ")"

<AttributeGroup> ::=
    ( <SingleAttribute> <Ws> )*
    ( <UseAttributeGroup> <Ws> )*

<UseAttributeGroup> ::= "USEATTRIBUTES" <Ws> <StringLiteral> <Ws>

<SingleObject> ::=
    <RouteObject> |
    <IdentifiableObject> |
    <ReferableObject> |
    <FragmentObject> |
    <DescriptorObject>
```

```
<RouteObject> ::= "ROUTE" <Ws> <RouteLiteral> <Ws>

<IdentifierInstance> ::= "(" <StringLiteral> ")"
<IdentifierInstanceOrAll> ::= <IdentifierInstance> | "(\"*\")"

<IdentifiableObject> ::=
    "IDENTIFIABLE" <Ws> ( "$aas" | "$sm" | "$cd" ) <IdentifierInstanceOrAll>

<ReferableObject> ::= "REFERABLE" <Ws> "$sme" <IdentifierInstanceOrAll> "." <IdShortPath>

<FragmentObject> ::= "FRAGMENT" <Ws> <FragmentLiteral> <Ws>

<DescriptorObject> ::= "DESCRIPTOR" <Ws> ( "$aasdesc" | "$smdesc" )
<IdentifierInstanceOrAll>

<ObjectGroup> ::=
    ( <SingleObject> <Ws> )* |
    ( <UseObjectGroup> <Ws> )*

<UseObjectGroup> ::= "USEOBJECTS" <Ws> <StringLiteral> <Ws>

<UseFormula> ::= "USEFORMULA" <Ws> <StringLiteral> <Ws>

<LogicalExpression> ::=
    <LogicalNestedExpression> |
    <LogicalOrExpression> |
    <LogicalAndExpression> |
    <LogicalNotExpression> |
    <MatchExpression> |
    <BoolLiteral> |
    <CastToBool> |
    <SingleComparison>

<LogicalNestedExpression> ::= "(" <Ws> <LogicalExpression> ")" <Ws>

<LogicalOrExpression> ::=
    "$or" <Ws> "(" <Ws> <LogicalExpression>
    ( "," <Ws> <LogicalExpression> )+ ")" <Ws>

<LogicalAndExpression> ::=
    "$and" <Ws> "(" <Ws> <LogicalExpression>
    ( "," <Ws> <LogicalExpression> )+ ")" <Ws>

<LogicalNotExpression> ::= "$not" <Ws> "(" <Ws> <LogicalExpression> ")" <Ws>

<MatchExpression> ::=
    "$match" <Ws> "(" <Ws>
    ( <SingleComparison> | <MatchExpression> )
    ( "," <Ws> ( <SingleComparison> | <MatchExpression> ) )*
    ")" <Ws>
```

```
<SingleComparison> ::=
    <StringComparison> |
    <NumericalComparison> |
    <HexComparison> |
    <BoolComparison> |
    <DateTimeComparison> |
    <TimeComparison>

<AllComparisons> ::= "$eq" | "$ne" | "$gt" | "$lt" | "$ge" | "$le"

<StringComparison> ::=
    (
        ( "$starts-with" | "$ends-with" | "$contains" | "$regex" )
        <Ws> "(" <Ws> <StringOperand> <Ws> "," <Ws> <StringOperand> <Ws> ")" <Ws>
    ) |
    ( <StringOperand> <Ws> <AllComparisons> <Ws> <StringNonFieldOperand> <Ws> ) |
    ( <StringNonFieldOperand> <Ws> <AllComparisons> <Ws> <StringOperand> <Ws> )

<NumericalComparison> ::=
    ( <NumericalOperand> <Ws> <AllComparisons> <Ws> <NumericalOperand> <Ws> ) |
    ( <NumericalOperand> <Ws> <AllComparisons> <Ws> <FieldIdentifier> <Ws> ) |
    ( <FieldIdentifier> <Ws> <AllComparisons> <Ws> <NumericalOperand> <Ws> )

<HexComparison> ::=
    ( <HexOperand> <Ws> <AllComparisons> <Ws> <HexOperand> <Ws> ) |
    ( <HexOperand> <Ws> <AllComparisons> <Ws> <FieldIdentifier> <Ws> ) |
    ( <FieldIdentifier> <Ws> <AllComparisons> <Ws> <HexOperand> <Ws> )

<BoolComparison> ::=
    ( <BoolOperand> <Ws> ( "$eq" | "$ne" ) <Ws> <BoolOperand> <Ws> ) |
    ( <BoolOperand> <Ws> ( "$eq" | "$ne" ) <Ws> <FieldIdentifier> <Ws> ) |
    ( <FieldIdentifier> <Ws> ( "$eq" | "$ne" ) <Ws> <BoolOperand> <Ws> )

<DateTimeComparison> ::=
    ( <DateTimeOperand> <Ws> <AllComparisons> <Ws> <DateTimeOperand> <Ws> ) |
    ( <DateTimeOperand> <Ws> <AllComparisons> <Ws> <FieldIdentifier> <Ws> ) |
    ( <FieldIdentifier> <Ws> <AllComparisons> <Ws> <DateTimeOperand> <Ws> )

<DateTimeToNum> ::=
    ( "$dayOfWeek" | "$dayOfMonth" | "$month" | "$year" )
    <Ws> "(" <Ws> <DateTimeOperand> <Ws> ")" <Ws>

<TimeComparison> ::=
    ( <TimeOperand> <Ws> <AllComparisons> <Ws> <TimeOperand> <Ws> ) |
    ( <TimeOperand> <Ws> <AllComparisons> <Ws> <FieldIdentifier> <Ws> ) |
    ( <FieldIdentifier> <Ws> <AllComparisons> <Ws> <TimeOperand> <Ws> )

<Operand> ::=
    <StringOperand> |
    <NumericalOperand> |
    <HexOperand> |
    <BoolOperand> |
```

```
    <DateTimeOperand> |
    <TimeOperand>

<StringOperand> ::= <FieldIdentifier> | <StringLiteral> | <CastToString> |
<SingleAttribute>

<StringNonFieldOperand> ::= <StringLiteral> | <CastToString> | <SingleAttribute>

<NumericalOperand> ::= <NumericalLiteral> | <CastToNumerical> | <DateTimeToNum>

<HexOperand> ::= <HexLiteral> | <CastToHex>

<BoolOperand> ::= <BoolLiteral> | <CastToBool>

<DateTimeOperand> ::= <DateTimeLiteral> | <CastToDateTime> | <DateTimeGlobalAttribute>

<TimeOperand> ::= <TimeLiteral> | <CastToTime>

<CastToString> ::= "str" <Ws> "(" <Ws> <Operand> <Ws> ")" <Ws>

<CastToNumerical> ::= "num" <Ws> "(" <Ws> <Operand> <Ws> ")" <Ws>

<CastToHex> ::= "hex" <Ws> "(" <Ws> <Operand> <Ws> ")" <Ws>

<CastToBool> ::= "bool" <Ws> "(" <Ws> <Operand> <Ws> ")" <Ws>

<CastToDateTime> ::= "dateTime" <Ws> "(" <Ws> <StringOperand> <Ws> ")" <Ws>

<CastToTime> ::= "time" <Ws> "(" <Ws> ( <StringOperand> | <DateTimeOperand> ) <Ws> ")" <Ws>

<DateTimeLiteral> ::= <Datetime> <Ws>
<TimeLiteral> ::= <Time> ( <Timezone> )? <Ws>
<Datetime> ::= <Date> "T" <Time> ( <Timezone> )?
<Date> ::= <Year> "-" <Month> "-" <Day>
<Year> ::=
    ( "-" )?
    ( ( [1-9] <Digit> <Digit> <Digit> <Digit>* ) | ( "0" <Digit> <Digit> <Digit> ) )
<Month> ::= ( "0" [1-9] ) | ( "1" [0-2] )
<Day> ::= ( "0" [1-9] ) | ( ( "1" | "2" ) <Digit> ) | ( "3" ( "0" | "1" ) )
<Time> ::= <NormalTime> | <EndOfDayTime>
<NormalTime> ::= <Hour> ":" <Minute> ":" <Second> ( "." <Fraction> )?
<EndOfDayTime> ::= "24:00:00" ( "." "0"+ )?
<Timezone> ::=
    (
        "Z" |
        ( "+" | "-" ) ( ( ( "0" <Digit> | "1" [0-3] ) ":" <Minute> ) | "14:00" )
    )
<Hour> ::= ( "0" | "1" ) <Digit> | "2" [0-3]
<Minute> ::= [0-5] <Digit>
<Second> ::= [0-5] <Digit>
<Fraction> ::= <Digit>+
```

```
<Digit> ::= [0-9]
<StringLiteral> ::=
    "\""
    (
        [A-Z] | [a-z] | [0-9] | "/" | "*" | "[" | "]" | "(" | ")" |
        " " | "_" | "@" | "#" | "\\" | "+" | "-" | "." | "," | ":" | "$" | "^"
    )+
    "\""
<ClaimLiteral> ::= <StringLiteral>
<RouteLiteral> ::= <StringLiteral>
<IdentifiableLiteral> ::= <StringLiteral>
<ReferableLiteral> ::= <StringLiteral>
<FragmentLiteral> ::= <StringLiteral>
<DescriptorLiteral> ::= <StringLiteral>
<NumericalLiteral> ::=
    ( "+" | "-" )?
    ( [0-9]+ ( "." [0-9]* )? | "." [0-9]+ )
    ( ( "e" | "E" ) ( "+" | "-" )? [0-9]+ )?
<HexLiteral> ::= "16#" ( [0-9] | [A-F] )+
<BoolLiteral> ::= "true" | "false"

<FieldIdentifier> ::=
    <FieldIdentifierAAS> |
    <FieldIdentifierSM> |
    <FieldIdentifierSME> |
    <FieldIdentifierCD> |
    <FieldIdentifierAasDescriptor> |
    <FieldIdentifierSmDescriptor>

<FieldIdentifierAAS> ::= "$aas#" <FieldsAAS>
<FieldIdentifierSM> ::= "$sm#" <FieldsSM>
<FieldIdentifierSME> ::= "$sme" ( "." <IdShortPath> )? "#" <FieldsSME>
<FieldIdentifierCD> ::= "$cd#" <FieldsCD>
<FieldIdentifierAasDescriptor> ::= "$aasdesc#" <FieldsAasDescriptor>
<FieldIdentifierSmDescriptor> ::= "$smdesc#" <SmDescriptorClause>

<FieldsAAS> ::=
    "idShort" |
    "id" |
    "assetInformation.assetKind" |
    "assetInformation.assetType" |
    "assetInformation.globalAssetId" |
    "assetInformation." <SpecificAssetIdsClause> |
    "submodels" ( "[" <ArrayIndex>? "]" ) ( "." <ReferenceClause> )?

<FieldsSM> ::= <SemanticIdClause> | <SupplementalSemanticIdClause> | "idShort" | "id"

<FieldsSME> ::=
    <SemanticIdClause> |
    <SupplementalSemanticIdClause> |
    "idShort" |
    "value" |
```

```
    "valueType" |
    "language"

<FieldsCD> ::= "idShort" | "id"

<FieldsAasDescriptor> ::=
    "idShort" |
    "id" |
    "assetKind" |
    "assetType" |
    "globalAssetId" |
    <SpecificAssetIdsClause> |
    "endpoints" ( "[" <ArrayIndex>? "]" ) "." <EndpointClause> |
    "submodelDescriptors" ( "[" <ArrayIndex>? "]" ) "." <SmDescriptorClause>

<SmDescriptorClause> ::=
    <SemanticIdClause> |
    <SupplementalSemanticIdClause> |
    "idShort" |
    "id" |
    "endpoints" ( "[" <ArrayIndex>? "]" ) "." <EndpointClause>

<EndpointClause> ::= "interface" | "protocolinformation.href"

<ReferenceClause> ::= "type" | "keys" ( "[" <ArrayIndex>? "]" ) ( ".type" | ".value" )
<SemanticIdClause> ::= "semanticId" | "semanticId." <ReferenceClause>
<SupplementalSemanticIdClause> ::= "supplementalSemanticIds" ( "[" <ArrayIndex>? "]" )? (
"." <ReferenceClause> )?
<SpecificAssetIdsClause> ::=
    "specificAssetIds" ( "[" <ArrayIndex>? "]" )
    (
        ".name" |
        ".value" |
        ".externalSubjectId" |
        ".externalSubjectId." <ReferenceClause>
    )

<FieldIdentifierFragment> ::=
    <FieldIdentifierAASFragment> |
    <FieldIdentifierSMFragment> |
    <FieldIdentifierSMEFragment> |
    <FieldIdentifierCDFragment> |
    <FieldIdentifierAasDescriptorFragment> |
    <FieldIdentifierSmDescriptorFragment>

<FieldIdentifierAASFragment> ::=
    "$aas#"
    (
        "idShort" |
        "assetInformation.assetType" |
        "assetInformation.globalAssetId" |
        "assetInformation." <SpecificAssetIdsClauseFragment> |
```

```
        <SubmodelsClauseFragment>
    )

<FieldIdentifierSMFragment> ::=
    "$sm#" ( <SemanticIdClauseFragment> | <SupplementalSemanticIdClauseFragment> |
"idShort" )

<FieldIdentifierSMEFragment> ::=
    "$sme" ( "." <IdShortPath> )?
    (
        "#"
        (
            <SemanticIdClauseFragment> |
            <SupplementalSemanticIdClauseFragment> |
            "idShort" |
            "value" |
            "valueType" |
            "language"
        )
    )?

<FieldIdentifierCDFragment> ::= "$cd#" "idShort"

<FieldIdentifierAasDescriptorFragment> ::=
    "$aasdesc#"
    (
        "idShort" |
        "description" |
        "displayName" |
        "extension" |
        "administration" |
        "assetKind" |
        "assetType" |
        "globalAssetId" |
        <SpecificAssetIdsClauseFragment> |
        <EndpointClauseFragment> |
        <SubmodelDescriptorsClauseFragment>
    )

<FieldIdentifierSmDescriptorFragment> ::= "$smdesc#" <SmDescriptorClauseFragment>

<SpecificAssetIdsClauseFragment> ::=
    "specificAssetIds" |
    "specificAssetIds" ( "[" <ArrayIndex>? "]" )
    ( ".externalSubjectId" ( "." <ReferenceClauseFragment> )? )?

<SubmodelsClauseFragment> ::=
    "submodels" |
    "submodels" ( "[" <ArrayIndex>? "]" ) ( "." <ReferenceClauseFragment> )?

<SubmodelDescriptorsClauseFragment> ::=
    "submodelDescriptors" |
```

```
    "submodelDescriptors" ( "[" <ArrayIndex>? "]" ) ( "." <SmDescriptorClauseFragment> )?

<SmDescriptorClauseFragment> ::=
    <SemanticIdClauseFragment> |
    <SupplementalSemanticIdClauseFragment> |
    "idShort" |
    <EndpointClauseFragment>

<EndpointClauseFragment> ::= "endpoints" ( "[" <ArrayIndex>? "]" )?

<SemanticIdClauseFragment> ::= "semanticId" | "semanticId." <ReferenceClauseFragment>

<SupplementalSemanticIdClauseFragment> ::=
    "supplementalSemanticIds" ( "[" <ArrayIndex>? "]" ( "." <ReferenceClauseFragment> )? )?

<ReferenceClauseFragment> ::= "keys" ( "[" <ArrayIndex>? "]" )?

<IdShortPath> ::= ( <IdShort> ( "[" <ArrayIndex>? "]" )* ( "." <IdShortPath> )* )
<IdShort> ::=
    (
        ( [a-z] | [A-Z] )
        ( [a-z] | [A-Z] | [0-9] | "_" | "-" )*
        ( [a-z] | [A-Z] | [0-9] | "_" )
    )
<ArrayIndex> ::= "0" | ( [1-9] [0-9]* )

<Ws> ::= ( " " | "\t" | "\r" | "\n" )*
```

Explanation of the Access Rule Model

General

The AAS Access Rule Model can be used to describe access rules. Whether and how access rules are enforced is beyond the specification of the model for access control. The parties involved are supposed to agree on governance and policies.

The AAS Access Rule Model uses Attribute Based Access Control (ABAC), i.e. Attributes are used in access rules. By ABAC also Role Based Access Control (RBAC) can be implemented by defining role attributes. Subject Attributes and Roles may be provided as claims in Access Tokens.

Attributes in access rules are either claims from an Access Token provided by an Identity Provider, global attributes like actual DATETIME or from a Submodel like a property for a machine state.

Objects to be protected are either API Routes, Identifiables (e.g. AAS or Submodel), Referables (e.g. SubmodelElements), Descriptors or Fragments of all those (e.g. AssetId, SemanticId, SpecificAssetId).

Access Rule Model

An Access Rule Model <AllAccessPermissionRules> defines a list of Access Permission Rules.

The access rule model <AllAccessPermissionRules> also contains lists of Attribute Groups <DEFATTRIBUTES>, Object Groups <DEFOBJECTS>, ACLs <DEFACLS> and Formulas <DEFFORMULAS>. These pre-defines can be referred to from other parts of the access control rules by the assigned <StringLiteral>. This allows to reuse commonly used definitions instead of having multiple copies spread over the access control rules. See Reusable Definitions.

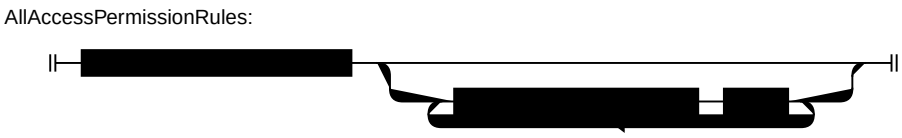


Figure 16. Railroad diagram for <AllAccessPermissionRules>

Reusable Definitions

The AAS Access Rule Model allows to define modular parts which can be reused in different access rules.

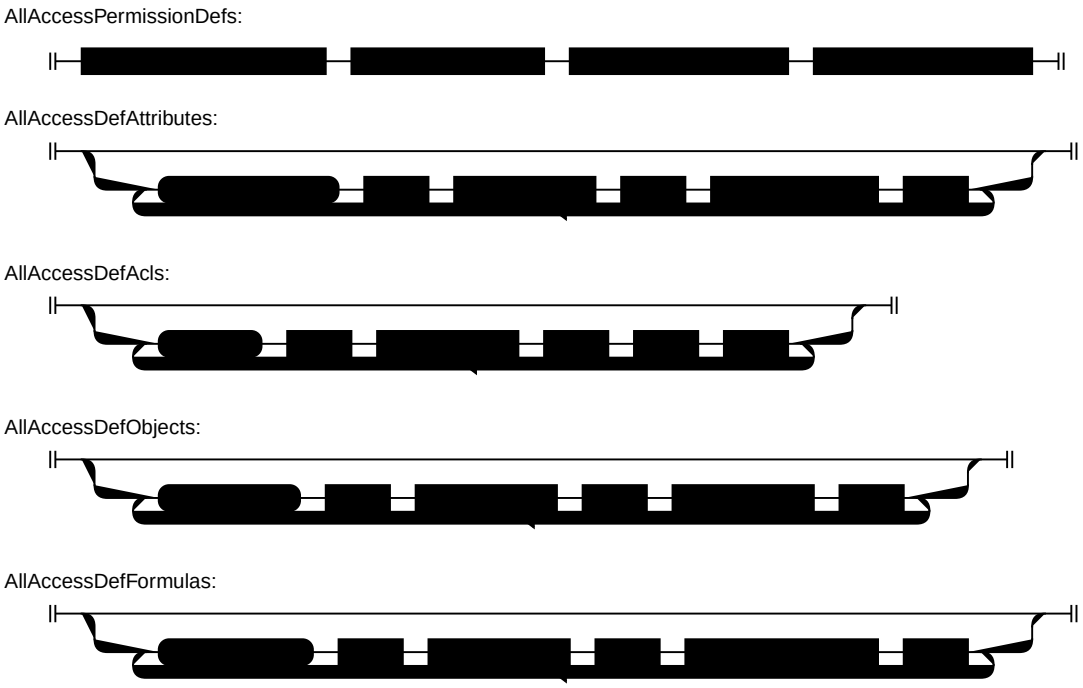


Figure 17. Railroad diagram for reusable definitions

Table Table 3 summarizes the details for pre-definitions used in <AllAccessPermissionRules>.

Table 3. <AllAccessPermissionRules> pre-defines

Definition	Description
DEFATTRIBUTES	Pre-defined <AttributeGroup>. Attributes can be combined into related groups, which may also be used in other groups. The group can be referenced using <UseAttributeGroup>.
DEFOBJECTS	Pre-defined <ObjectGroup>. Objects can be combined into related groups, which may also be used in other groups. The group can be referenced using <UseObjectGroup>.
DEFACLS	Pre-defined <ACL>. An ACL defines which access rights are given for a certain combination of attributes. The ACL can be referenced using <UseACL>.
DEFFORMULAS	Pre-defined formula. Formulas can be referenced using <UseFormula>.

EXAMPLE See Reuse of ACL, OBJECT and FORMULA.

Access Permission Rules

<AccessPermissionRule> defines a single access control rule granting permissions on the designated AAS objects.

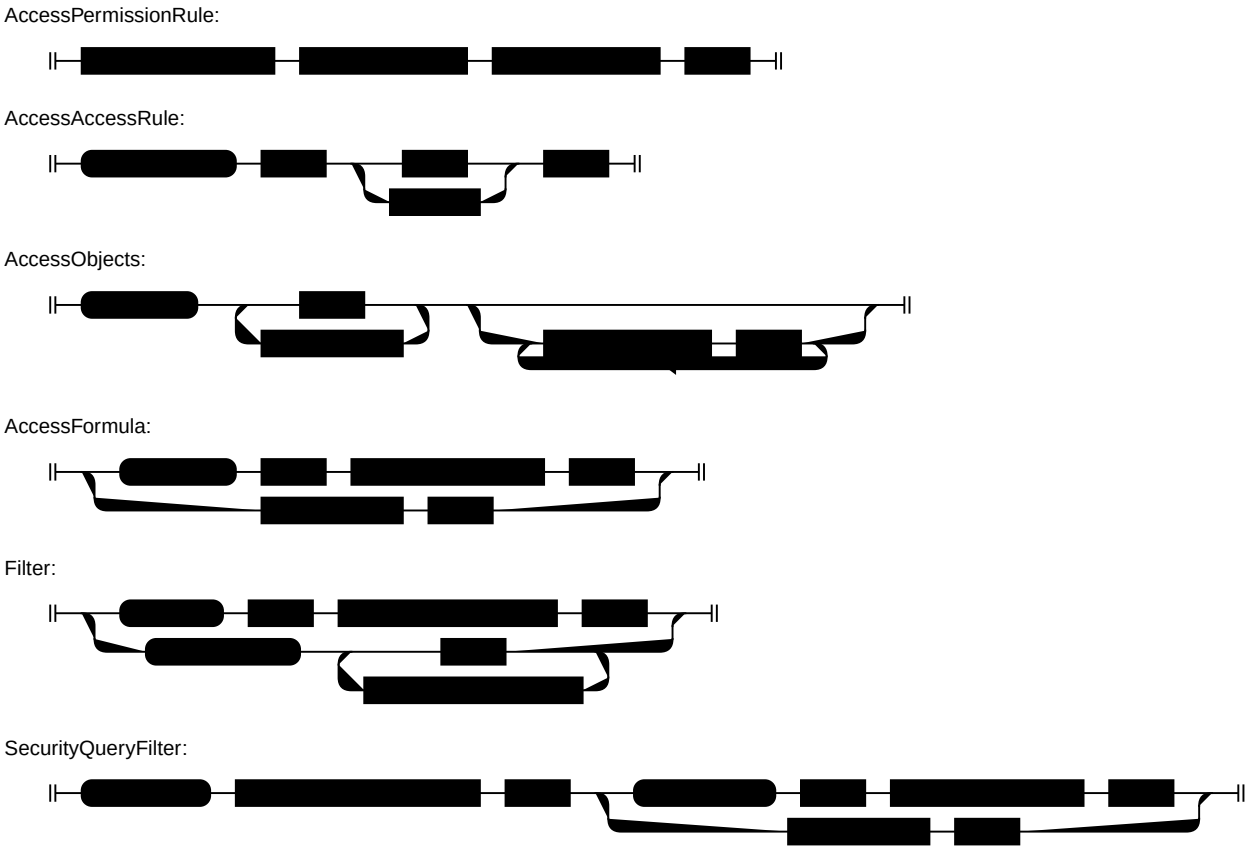


Figure 18. Railroad diagram for <AccessPermissionRule>

Table Table 4 describes the details of <AccessPermissionRule>.

Table 4. <AccessPermissionRule>

Section	Description
ACCESSRULE:	Either specifies the access control list (<ACL>) or contains a reference to an access control list (<UseACL>) pre-defined using DEFACLS. Access control lists are described in ACL.
OBJECTS:	Defines the set of AAS objects to which the access permission applies. The list contains individual objects and/or refers to object groups pre-defined using DEFOBJECTS. Objects are described in AAS Objects.
FORMULA:	Defines the formula to be evaluated to determine whether the access permission is in effect. Predefined formulas can be referenced using <UseFormula>. Formulas allow only expressions with Boolean results, e.g., comparisons; arithmetic is currently not supported. If the formula evaluates to true the access permission shall be enforced. Formulas are described in Conditions and Formulas.

Section	Description
FILTER: FILTERLIST:	Filters can be used to restrict the exposure of an AAS object depending on the object values. Only values passing through the filter are returned to the AAS user application. Any AAS object with values that do not match the filter expression shall be removed from the list of returned objects. EXAMPLE See Example with FILTER statement.

An access rule may optionally include <FILTER> or <FILTERLIST> to further restrict the returned objects. Without a filter, access is granted to the complete object covered by the access rule.

A filter is represented by <SecurityQueryFilter>. It combines a <FRAGMENT> with either a <CONDITION> or a reference to a pre-defined formula using <UseFormula>. The <CONDITION> is a Boolean <logicalExpression> that defines whether the referenced fragment of the given object can be accessed. <FILTER> contains one <SecurityQueryFilter>. <FILTERLIST> contains a list of <SecurityQueryFilter> elements and can therefore define multiple fragment-level restrictions.

The <FRAGMENT> identifies which part of the accessed object has to be filtered. The related <FragmentLiteral> defines the prefix of a <FieldIdentifier> to be filtered, e.g. "$aasdesc#specificAssetIds[]" defines that the specificAssetIds part shall be filtered.

For a filter addressing SubmodelElements (Fragment $sme), the filter condition SHALL be evaluated separately for each SubmodelElement identified by its ID.

Visibility is applied recursively to the element hierarchy: a SubmodelElement SHALL be returned only if it and all its parent elements are visible.

ACL

An ACL (Access Control List) defines which access rights apply depending on a required set of attributes.

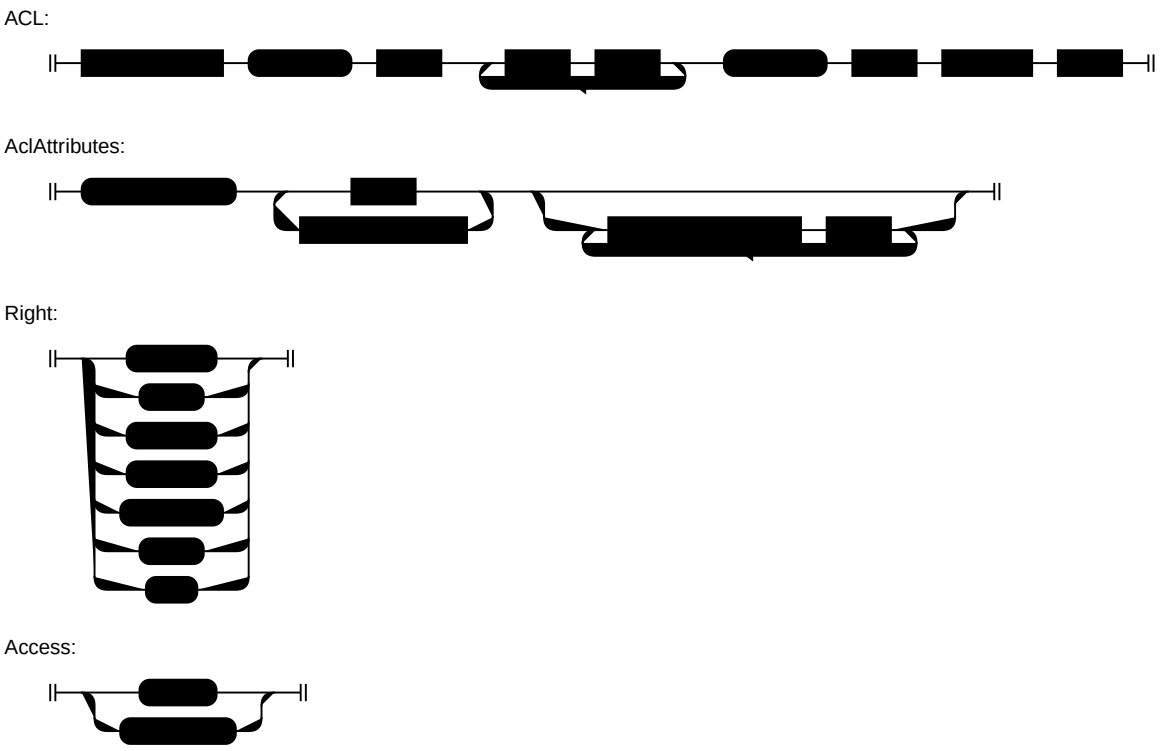


Figure 19. Railroad diagram for <ACL>

Table Table 5 describes the details of <ACL>.

Table 5. <ACL>

Section	Description
ATTRIBUTES:	List of ABAC attributes that the access control list depends on. The list contains definitions for individual attributes or pre-defined attribute groups introduced with DEFATTRIBUTES. Attributes are described in Attributes.
RIGHTS:	Enumeration of access rights subject to this access control list. Table Table 7 describes the possible access rights, their semantics, and how they apply to the interface operations defined by the AAS API [2].
ACCESS:	Type of access managed by this access control list. See Table Table 6 for the description of possible access types.

Table 6. <Access>

Access type enumeration	Description
"ALLOW"	The respective permission is granted if the associated formula evaluates to true (see Conditions and Formulas). Without an ALLOW rule any access is forbidden by default.
"DISABLED"	This access control list shall not be evaluated and is ignored. This can be used for test purposes.

USEACL can be used to refer to a predefined ACL (see Reusable Definitions).

Figure 20. Railroad diagram for <UseACL>

Rights and operation verbs

The RIGHTS values are technology-neutral permissions. For HTTP/REST APIs these permissions are typically enforced on interface operations and their mapped HTTP methods. The mapping in Table Table 7 is intended to improve interoperability when ROUTE objects are used for authorization decisions. Concrete informative examples for RIGHTS, ROUTEs and HTTP requests are given in Examples of ROUTEs and HTTP requests.

The normative operation-level mapping for the AAS HTTP/REST API (per-operation RIGHT and ROUTE) is defined in IDTA-01002 § "Operation to RIGHT Mapping" (annex). Implementations of this specification MUST follow that mapping when evaluating access rules against concrete HTTP operations.

Table 7. Indicative mapping of RIGHTS to AAS operation verbs and HTTP methods

Right	Operation verb	HTTP method	Meaning
CREATE	Post	POST	Permission to create new AAS objects where the parent object is the AAS object designated in the OBJECTS: section of the Access Permission Rule (Table Table 4).

Right	Operation verb	HTTP method	Meaning
CREATE	Put	PUT	Permission to create new AAS objects where the parent object is the AAS object designated in the OBJECTS: section of the Access Permission Rule (Table Table 4).
READ	Get, GetAll	GET	Permission to perform read operations on the AAS objects designated in the OBJECTS: section of the Access Permission Rule (Table Table 4). FILTER: can be used to filter the result data on a sub-object granularity from the object data returned by the Get/GetAll operation. EXAMPLE See Example with FILTER statement.
UPDATE	Patch	PATCH	Permission to perform update operations on existing the AAS objects designated in the OBJECTS: section of the Access Permission Rule (Table Table 4).
UPDATE	Put	PUT	Permission to replace an existing AAS object designated in the OBJECTS: section of the Access Permission Rule (Table Table 4).
DELETE	Delete	DELETE	Permission to perform delete operations on the AAS objects designated in the OBJECTS: section of the Access Permission Rule (Table Table 4).
EXECUTE	Invoke	POST	Permission to invoke an operation.
VIEW	GetAll	GET	View the existence of an AAS object as ID or idShort. This does not include access to the value and any other attributes of the AAS element. EXAMPLE It is allowed to see, that a specific Submodel is present, but this does not imply, that access to any information stored within the Submodel is granted.
ALL	n/a	n/a	Permission for full access on the designated AAS objects, i.e., shorthand for a combination of all eligible permissions.

The mapping is not intended to redefine AAS Part 2 operations.

It clarifies how the abstract RIGHT values are typically applied when an authorization decision is made for a concrete HTTP route. The same HTTP method can therefore be used for different rights depending on the targeted interface operation and the representation that is returned. For example, GET may correspond to READ for full content access or VIEW for reference-only or existence-only access. Search-style read operations may also use POST in specific interfaces. For endpoints with create-or-replace or upsert semantics, the required right depends on whether the addressed resource already exists.

For example, PUT /api/v3/submodels/{submodelIdentifier} requires CREATE if the addressed submodel does not yet exist and UPDATE if it already exists. The same principle applies to endpoint-specific POST operations such as POST

/api/v3/lookup/shells/{aasIdentifier} if the endpoint creates the addressed resource when absent and updates it when present.

Attributes

<SingleAttribute> describes an ABAC attribute which is to be used in an ACL (Table Table 5).

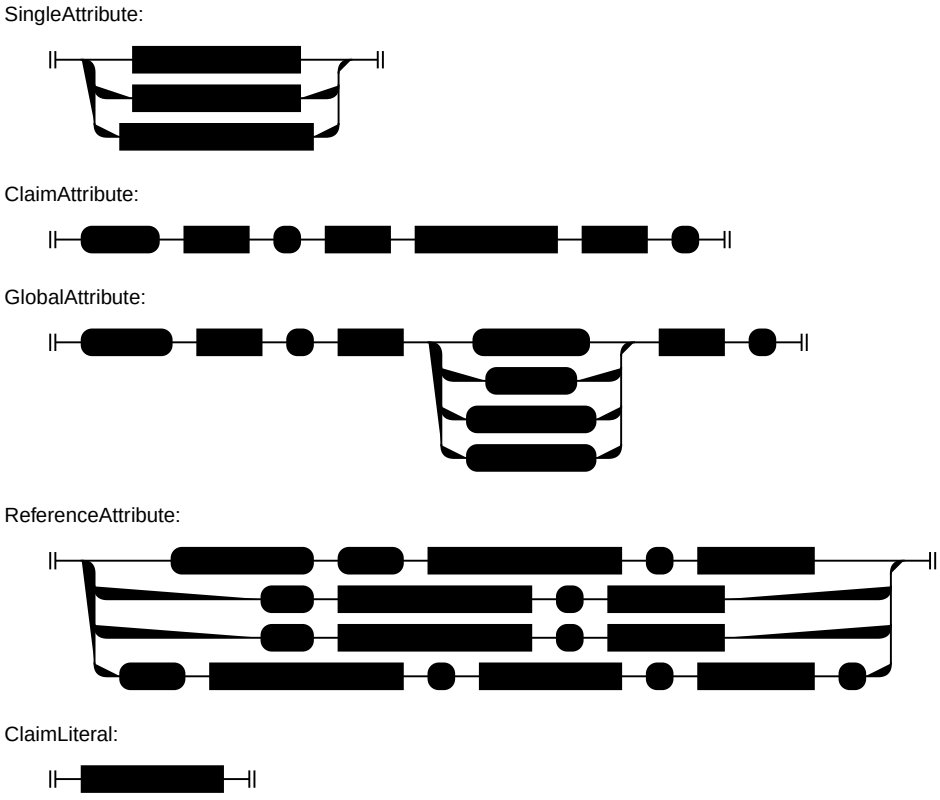


Figure 21. Railroad diagram for attributes

Table Table 8 describes the available ABAC attribute types.

Table 8. <ClaimAttribute>, <GlobalAttribute> and <ReferenceAttribute>

Attribute enumeration	Description
"CLAIM"	ABAC attribute of the subject/requestor provided as a claim in the access token to the AAS interface (see Access token). EXAMPLE See authenticated users and BusinessPartnerNumber.
"GLOBAL"	Global/environment condition available to the AAS interface, e.g., time.
"REFERENCE"	Reference to an attribute of any Referable as defined by the AAS metamodel [1] within the scope of the AAS accessed via the AAS interface.

Single Attributes are either claims from an Access Token provided by an Identity Provider, global attributes like actual DATETIME or references to a model field, e.g. to a SubmodelElement property for a machine state. An Attribute Group defines a list of single attributes and/or a list of names of other attribute groups.

A REFERENCE attribute addresses a field by identifier, independently of the object currently evaluated. This is useful when a rule for one object needs state data from another model context, for example checking a machine state from an OperationalData Submodel while filtering maintenance documents in a different Submodel. EXAMPLE See

Reference Attribute and state-dependent filtering.

Table Table 9 gives the details for global attributes.

Table 9. <GlobalAttribute>

Global attribute enumeration	Description
"LOCALNOW"	Local time of the AAS interface that shall be used when evaluating the access control rule. LOCALNOW shall be derived from a trustworthy time source. LOCALNOW shall include the information required to map to a valid <TimeLiteral>. NOTE This is not necessarily the local system time of the system evaluating the access control rule. The AAS Responsible decides what is used as local time by the AAS interface. EXAMPLE An AAS Responsible for a company located in central Europe may set LOCALNOW to central European time (CET) even if the AAS is hosted in Atlantic daylight time (ADT).
"UTCNOW"	Coordinated Universal Time. UTCNOW shall be derived from a trustworthy time source. UTCNOW shall include the information required to map to a valid <TimeLiteral>.
"CLIENTNOW"	Time as perceived by the AAS user application. A trustworthy value for CLIENTNOW can be established as result of the authentication process (see Authentication Flows) if sufficient information on the location of the subject is available. The actual value may then be provided as authenticated attribute of the access token (see Access token). CLIENTNOW shall include the information required to map to a valid <TimeLiteral>.
"ANONYMOUS"	Anonymous access, i.e., the request does not contain an access token. ANONYMOUS is intended to be used for access control rules permitting access to public information provided by the AAS. EXAMPLE See complete API and list of semanticIDs.

<AttributeGroup> allows to define a set of attributes in a pre-defined group (see Reusable Definitions) that can be referenced by USEATTRIBUTES. Attribute groups may be recursive, i.e., an attribute group can be composed of attribute groups.

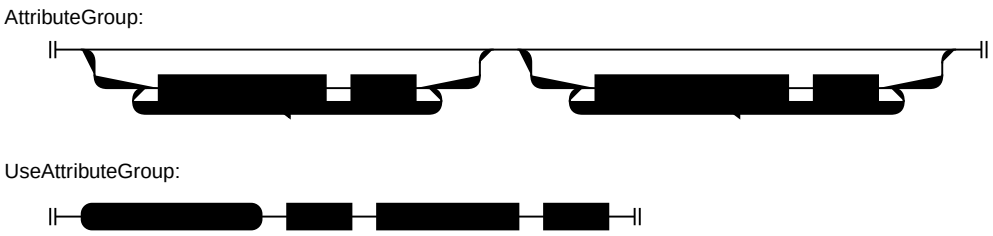


Figure 22. Railroad diagram for <AttributeGroup> and <UseAttributeGroup>

In order to use attributes as operands the query language defined in IDTA-01002, Query Language is extended.

The capability to use attributes as operands in the evaluation of access control rules is an important concept of ABAC. At the time of evaluation, the operand will be substituted with the actual value of the attributed, e.g., as provided within a claim of the access token.

The required additions for <StringOperand> and <StringNonFieldOperand> are shown in the BNF grammar.

AAS Objects

AAS objects are objects constituting the data represented by the AAS or an endpoint for an AAS operation (<RouteObject>).

<SingleObject> describes the AAS element to which an access control rule applies (see Table Table 4).

Objects to be protected are either API Routes, Identifiables (e.g. AAS or Submodel), Referables (e.g. SubmodelElements), Descriptors or Fragments of those (e.g. AssetId, SemanticId, SpecificAssetId). Table Table 10 explains the different options to refer to an AAS element.

Details on identifiers used by <IdentifiableObject>, <ReferableObject>, <FragmentObject>, and <DescriptorObject> are described in Identifiers.

Figure 23. Railroad diagram for objects

Table 10. <SingleObject>

Object type enumeration	Description
"ROUTE"	A route defines the endpoint address of an interface operation as defined by the AAS API [2]. This allows to define access control for requested interface operations, interface operation groups, or complete interfaces. An URL path (IETF RFC 1738, 3.1) preceded by “/” can be used with AAS to specify routes. An asterisk (“*”) can be used at the end of a <RouteLiteral> to refer to interface operation groups or complete interfaces. EXAMPLE See example requests by right.
"IDENTIFIABLE"	Reference to an Identifiable; text string as defined by the AAS API [2].
"REFERABLE"	Reference to a Referable; text string as defined by the AAS API [2].
"FRAGMENT"	Reference to a fragment; text string as defined by the AAS API [2].
"DESCRIPTOR"	A descriptor object, e.g. an AssetAdministrationShellDescriptor or a Submodel Descriptor as defined by the AAS API [2].

Details on identifiers used by <IdentifiableObject>, <ReferableObject>, <FragmentObject>, and <DescriptorObject> are described in Identifiers.

ROUTE defines the endpoint address of an interface operation. This allows access control to be defined for single interface operations, groups of interface operations, or complete interfaces.

For AAS HTTP/REST APIs, a ROUTE literal shall use the URL path part only, i.e. the part of the endpoint address starting with "/". The scheme, authority, query part and fragment identifier are not part of ROUTE matching. The deployment-specific server base URL is therefore handled separately from the ROUTE literal.

ROUTE matching uses the following rules:

- A ROUTE literal starting with "/" and not ending with "*" matches one concrete path.
- The literal "*" matches all routes.
- A ROUTE literal ending with "*" matches all routes with the given prefix.

Examples are "/shells", "/shells/**" and "**". For HTTP/REST APIs the examples in Examples of ROUTEs and HTTP requests show how ROUTE literals can be combined with RIGHTS and concrete request examples from AAS Part 2.

<ObjectGroup> allows to combine a set of objects in pre-defined group (see Reusable Definitions) that can be referenced by USEOBJECTS. Object groups may be recursive, i.e., an object group can be composed of other object groups.

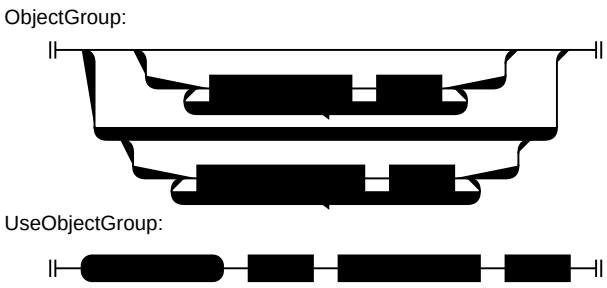


Figure 24. Railroad diagram for <ObjectGroup> and <UseObjectGroup>

Identifiers

References to attribute values (see Attributes) and objects (see AAS Objects) use instance and field identifiers.

Instances can be referred to by a specific identifier or a wildcard “*” to apply to all objects of the specified type. The syntax of the <StringLiteral> used to identify a specific instance shall be defined by the AAS Responsible according to the identifier policy described in Identifier management.

Table Table 11 enumerates the available identifiers.

Table 11. Field Identifiers

Field identifier	Description
$aas#	Prefix for value of AAS attribute fields
$sm#	Prefix for value of Submodel attribute fields
$sme#	Prefix for value of SubmodelElement attribute fields
$cd#	Prefix for value of Concept Description attribute fields
$aasdesc#	Prefix for value of AAS descriptor attribute fields. AAS descriptors are the entries in AAS registries.
$smdesc#	Prefix for value of Submodel descriptor attribute fields. Submodel descriptors are the entries in Submodel registries.

Conditions and Formulas

Any <LogicalExpression> described using the AAS query language defined by IDTA-01002, Query Language may be used in a formula for access control.

The logical expression used as condition shall evaluate to either true or false. If the expression evaluates to true the respective Access Permission Rule shall be enforced, otherwise the rule is ignored.

Arithmetic in Formulas is currently not supported.

Nested logical expressions create combinations of several logical expressions, where any, all or none of the expressions needs to be enabled: **<logicalNestedExpression>, <logicalOrExpression>, <logicalAndExpression>, <logicalNotExpression>**.

String Comparison Operations compare or match the first given argument (left argument) with the second given argument (right argument). **$eq, $ne, $gt, $lt, $ge, $le** make an alphabetic string comparison. $**starts-with, $ends-with, $contains and $regex** check, if the first given argument is part of the second argument or if the first argument matches with the given REGEX.

Numerical Comparison Operations compare the first given argument (left argument) with the second given argument (right argument). Since AAS also supports XS Datatypes Hex, Bool, DateTime and Time, related comparisons are available accordingly.

For specific comparisons datatypes can be casted to the other datatypes. If a FieldIdentifier is used directly in a typed comparison, it is implicitly interpreted as the comparison type. This implicit conversion is tried for all values addressed by the FieldIdentifier. Values that cannot be converted are ignored for this FieldIdentifier evaluation. If no addressed value can be converted, the comparison evaluates to FALSE.

Specific operations exist to extract parts from DateTime, i.e. $dayOfWeek, $dayOfMonth, $month, $year. This enables access rules related to week days or specific times in the year. EXAMPLE See weekday and time-based access.

An important special operation is $match, which can be used with any element containing a list of elements, e.g. semanticId[], specificAssetId[], SubmodelElementList or SubmodelElementCollection. The list element is written with [] to express, that $match shall check if a certain expression is true for at least one element in the list. FILTER defines one fragment restriction for a returned object, while FILTERLIST defines several fragment restrictions with independent conditions.

A formula can use all ABAC attributes defined in the ATTRIBUTES: section of the ACL. A formula referring to undefined attributes shall result in false. Any type of error during the evaluation of a formula shall be treated as if the formula resulted in false.

USEFORMULAS can be used to refer to a predefined formula (see Reusable Definitions).

Figure 25. Railroad diagram for <UseFormula>

JSON Serialization of Access Rule Model

The AAS Access Rule model can also be defined as a JSON schema. Since the related JSON schema shall also allow automatic code generation, specific constraints must be fulfilled by such JSON schema. The use of "oneof" is limited and type information for objects must be available.

It shall also be possible to check a JSON with the JSON schema.

Examples can be found in Examples of Access Rules in JSON serialization. The JSON Schema artifacts and all JSON examples included in the specification are tested in the GitHub Actions CI/CD workflow "Validate spec artifacts". This workflow runs tools/validate_spec_artifacts.py with the Python jsonschema package. The script checks the JSON Schema files as JSON Schema Draft 7, resolves their $ref references, runs schema smoke tests for selected definitions, and validates every JSON example from the specification against the common access-rule JSON Schema and the access-rule-model JSON Schema.

The following schema meets such constraints:

```
{
  "$schema": "http://json-schema.org/draft-07/schema#",
  "title": "Common JSON Schema for AAS Queries and Access Rules",
  "description": "This schema contains the AAS Access Rule Language.",
  "type": "object",
  "properties": {
    "AllAccessPermissionRules": {
      "$ref": "#/definitions/AllAccessPermissionRules"
    }
  },
  "required": [
    "AllAccessPermissionRules"
  ],
  "definitions": {
    "standardString": {
      "type": "string",
      "pattern": "^[A-Za-z0-9/\\*\\[\\]\\(\\) _@#\\\\+\\-\\.,:\\$\\^]+$"
    },
    "ReferenceIdentifier": {
```

```
      "type": "string",
      "pattern": "^(?:\\$aas\\(\"[A-Za-z0-9/\\*\\[\\]\\(\\) _@#\\\\+\\-
\\.,:\\$\\^]+\"\\)#(?:idShort|id|assetInformation\\.assetKind|assetInformation\\.assetType|
assetInformation\\.globalAssetId|assetInformation\\.specificAssetIds\\[(?:0|[1-9][0-
9]*)?\\]\\.(?:name|value|externalSubjectId(?:\\.(?:type|keys\\[(?:0|[1-9][0-
9]*)?\\]\\.(?:type|value)))?)|submodels\\[(?:0|[1-9][0-
9]*)?\\](?:\\.(?:type|keys\\[(?:0|[1-9][0-9]*)?\\]\\.(?:type|value)))?)|\\$sm\\(\"[A-Za-z0-
9/\\*\\[\\]\\(\\) _@#\\\\+\\-\\.,:\\$\\^]+\"\\)#(?:semanticId(?:\\.(?:type|keys\\[(?:0|[1-
9][0-9]*)?\\]\\.(?:type|value)))?|supplementalSemanticIds(?:\\[(?:0|[1-9][0-
9]*)?\\])?(?:\\.(?:type|keys\\[(?:0|[1-9][0-
9]*)?\\]\\.(?:type|value)))?|idShort|id)|\\$cd\\(\"[A-Za-z0-9/\\*\\[\\]\\(\\) _@#\\\\+\\-
\\.,:\\$\\^]+\"\\)#(?:idShort|id)|\\$sme\\(\"[A-Za-z0-9/\\*\\[\\]\\(\\) _@#\\\\+\\-
\\.,:\\$\\^]+\"\\)\\.[A-Za-z](?:[A-Za-z0-9_-]*[A-Za-z0-9_])?(?:\\[(?:0|[1-9][0-
9]*)?\\])*(?:\\.[A-Za-z](?:[A-Za-z0-9_-]*[A-Za-z0-9_])?(?:\\[(?:0|[1-9][0-
9]*)?\\])*)*#(?:semanticId(?:\\.(?:type|keys\\[(?:0|[1-9][0-
9]*)?\\]\\.(?:type|value)))?|supplementalSemanticIds(?:\\[(?:0|[1-9][0-
9]*)?\\])?(?:\\.(?:type|keys\\[(?:0|[1-9][0-
9]*)?\\]\\.(?:type|value)))?|idShort|value|valueType|language))$"
    },
    "IdentifiableIdentifier": {
      "type": "string",
      "pattern": "^\\$(?:aas|sm|cd)\\(\"[A-Za-z0-9/\\*\\[\\]\\(\\) _@#\\\\+\\-
\\.,:\\$\\^]+\"\\)$"
    },
    "ReferableIdentifier": {
      "type": "string",
      "pattern": "^\\$sme\\(\"[A-Za-z0-9/\\*\\[\\]\\(\\) _@#\\\\+\\-
\\.,:\\$\\^]+\"\\)\\.[A-Za-z](?:[A-Za-z0-9_-]*[A-Za-z0-9_])?(?:\\[(?:0|[1-9][0-
9]*)?\\])*(?:\\.[A-Za-z](?:[A-Za-z0-9_-]*[A-Za-z0-9_])?(?:\\[(?:0|[1-9][0-9]*)?\\])*)*$"
    },
    "FragmentIdentifier": {
      "$ref": "#/definitions/FragmentFieldIdentifier"
    },
    "DescriptorIdentifier": {
      "type": "string",
      "pattern": "^\\$(?:aasdesc|smdesc)\\(\"[A-Za-z0-9/\\*\\[\\]\\(\\) _@#\\\\+\\-
\\.,:\\$\\^]+\"\\)$"
    },
    "FieldIdentifier": {
      "type": "string",
      "pattern":
"^(?:\\$aas#(?:idShort|id|assetInformation\\.assetKind|assetInformation\\.assetType|assetIn
formation\\.globalAssetId|assetInformation\\.specificAssetIds\\[(?:0|[1-9][0-
9]*)?\\]\\.(?:name|value|externalSubjectId(?:\\.(?:type|keys\\[(?:0|[1-9][0-
9]*)?\\]\\.(?:type|value)))?)|submodels\\[(?:0|[1-9][0-
9]*)?\\](?:\\.(?:type|keys\\[(?:0|[1-9][0-
9]*)?\\]\\.(?:type|value)))?)|\\$sm#(?:semanticId(?:\\.(?:type|keys\\[(?:0|[1-9][0-
9]*)?\\]\\.(?:type|value)))?|supplementalSemanticIds(?:\\[(?:0|[1-9][0-
9]*)?\\])?(?:\\.(?:type|keys\\[(?:0|[1-9][0-
9]*)?\\]\\.(?:type|value)))?|idShort|id)|\\$sme(?:\\.[A-Za-z](?:[A-Za-z0-9_-]*[A-Za-z0-
9_])?(?:\\[(?:0|[1-9][0-9]*)?\\])*(?:\\.[A-Za-z](?:[A-Za-z0-9_-]*[A-Za-z0-
9_])?(?:\\[(?:0|[1-9][0-9]*)?\\])*)*)?#(?:semanticId(?:\\.(?:type|keys\\[(?:0|[1-9][0-
```

```
9]*)?\\]\\.(?:type|value)))?|supplementalSemanticIds(?:\\[(?:0|[1-9][0-
9]*)?\\])?(?:\\.(?:type|keys\\[(?:0|[1-9][0-
9]*)?\\]\\.(?:type|value)))?|idShort|value|valueType|language)|\\$cd#(?:idShort|id)|\\$aasd
esc#(?:idShort|id|assetKind|assetType|globalAssetId|specificAssetIds\\[(?:0|[1-9][0-
9]*)?\\]\\.(?:name|value|externalSubjectId(?:\\.(?:type|keys\\[(?:0|[1-9][0-
9]*)?\\]\\.(?:type|value)))?)|endpoints\\[(?:0|[1-9][0-
9]*)?\\]\\.(?:interface|protocolinformation\\.href)|submodelDescriptors\\[(?:0|[1-9][0-
9]*)?\\]\\.(?:semanticId(?:\\.(?:type|keys\\[(?:0|[1-9][0-
9]*)?\\]\\.(?:type|value)))?|supplementalSemanticIds(?:\\[(?:0|[1-9][0-
9]*)?\\])?(?:\\.(?:type|keys\\[(?:0|[1-9][0-
9]*)?\\]\\.(?:type|value)))?|idShort|id|endpoints\\[(?:0|[1-9][0-
9]*)?\\]\\.(?:interface|protocolinformation\\.href)))|\\$smdesc#(?:semanticId(?:\\.(?:type|
keys\\[(?:0|[1-9][0-9]*)?\\]\\.(?:type|value)))?|supplementalSemanticIds(?:\\[(?:0|[1-9][0-
9]*)?\\])?(?:\\.(?:type|keys\\[(?:0|[1-9][0-
9]*)?\\]\\.(?:type|value)))?|idShort|id|endpoints\\[(?:0|[1-9][0-
9]*)?\\]\\.(?:interface|protocolinformation\\.href)))$"
    },
    "FragmentFieldIdentifier": {
      "type": "string",
      "pattern":
"^(?:\\$aas#(?:idShort|assetInformation\\.assetType|assetInformation\\.globalAssetId|assetI
nformation\\.specificAssetIds(?:\\[(?:0|[1-9][0-
9]*)?\\](?:\\.externalSubjectId(?:\\.keys(?:\\[(?:0|[1-9][0-
9]*)?\\])?)?)?)?|submodels(?:\\[(?:0|[1-9][0-9]*)?\\](?:\\.keys(?:\\[(?:0|[1-9][0-
9]*)?\\])?)?)?)|\\$sm#(?:semanticId(?:\\.keys(?:\\[(?:0|[1-9][0-
9]*)?\\])?)?|supplementalSemanticIds(?:\\[(?:0|[1-9][0-9]*)?\\](?:\\.keys(?:\\[(?:0|[1-
9][0-9]*)?\\])?)?)?|idShort)|\\$sme(?:\\.[A-Za-z](?:[A-Za-z0-9_-]*[A-Za-z0-
9_])?(?:\\[(?:0|[1-9][0-9]*)?\\])*(?:\\.[A-Za-z](?:[A-Za-z0-9_-]*[A-Za-z0-
9_])?(?:\\[(?:0|[1-9][0-9]*)?\\])*)*)?(?:#(?:semanticId(?:\\.keys(?:\\[(?:0|[1-9][0-
9]*)?\\])?)?|supplementalSemanticIds(?:\\[(?:0|[1-9][0-9]*)?\\](?:\\.keys(?:\\[(?:0|[1-
9][0-
9]*)?\\])?)?)?|idShort|value|valueType|language))?|\\$cd#idShort|\\$aasdesc#(?:idShort|desc
ription|displayName|extension|administration|assetKind|assetType|globalAssetId|specificAsse
tIds(?:\\[(?:0|[1-9][0-9]*)?\\](?:\\.externalSubjectId(?:\\.keys(?:\\[(?:0|[1-9][0-
9]*)?\\])?)?)?)?|endpoints(?:\\[(?:0|[1-9][0-9]*)?\\])?|submodelDescriptors(?:\\[(?:0|[1-
9][0-9]*)?\\](?:\\.(?:semanticId(?:\\.keys(?:\\[(?:0|[1-9][0-
9]*)?\\])?)?|supplementalSemanticIds(?:\\[(?:0|[1-9][0-9]*)?\\](?:\\.keys(?:\\[(?:0|[1-
9][0-9]*)?\\])?)?)?|idShort|endpoints(?:\\[(?:0|[1-9][0-
9]*)?\\])?))?)?)|\\$smdesc#(?:semanticId(?:\\.keys(?:\\[(?:0|[1-9][0-
9]*)?\\])?)?|supplementalSemanticIds(?:\\[(?:0|[1-9][0-9]*)?\\](?:\\.keys(?:\\[(?:0|[1-
9][0-9]*)?\\])?)?)?|idShort|endpoints(?:\\[(?:0|[1-9][0-9]*)?\\])?))$"
    },
    "hexLiteralPattern": {
      "type": "string",
      "pattern": "^16#[0-9A-F]+$"
    },
    "dateTimeLiteralPattern": {
      "type": "string",
      "pattern": "^-?(([1-9][0-9][0-9][0-9]+)|(0[0-9][0-9][0-9]))-((0[1-9])|(1[0-2]))-
((0[1-9])|([12][0-9])|(3[01]))T(((([01][0-9])|(2[0-3])):[0-5][0-9]:([0-5][0-9])(\\.[0-
9]+)?)|24:00:00(\\.0+)?)(Z|(\\+|-)((0[0-9]|1[0-3]):[0-5][0-9]|14:00))?$"
    },
```

```
    "timeLiteralPattern": {
      "type": "string",
      "pattern": "^(([01][0-9]|2[0-3]):[0-5][0-9]:[0-5][0-9](\\.[0-
9]+)?|(24:00:00(\\.0+)?))(Z|(\\+|-)((0[0-9]|1[0-3]):[0-5][0-9]|14:00))?$"
    },
    "dateTimeAttributeItem": {
      "type": "object",
      "properties": {
        "GLOBAL": {
          "type": "string",
          "enum": [
            "LOCALNOW",
            "UTCNOW",
            "CLIENTNOW"
          ]
        }
      },
      "required": [
        "GLOBAL"
      ],
      "additionalProperties": false
    },
    "dateTimeOperand": {
      "type": "object",
      "properties": {
        "$dateTimeVal": {
          "$ref": "#/definitions/dateTimeLiteralPattern"
        },
        "$dateTimeCast": {
          "$ref": "#/definitions/stringValue"
        },
        "$attribute": {
          "$ref": "#/definitions/dateTimeAttributeItem"
        }
      },
      "oneOf": [
        {
          "required": [
            "$dateTimeVal"
          ]
        },
        {
          "required": [
            "$dateTimeCast"
          ]
        },
        {
          "required": [
            "$attribute"
          ]
        }
      ],
```

```
      "additionalProperties": false
    },
    "fieldOperand": {
      "type": "object",
      "properties": {
        "$field": {
          "$ref": "#/definitions/FieldIdentifier"
        }
      },
      "required": [
        "$field"
      ],
      "additionalProperties": false
    },
    "numericalOperand": {
      "type": "object",
      "properties": {
        "$numVal": {
          "type": "number"
        },
        "$numCast": {
          "$ref": "#/definitions/Value"
        },
        "$dayOfWeek": {
          "$ref": "#/definitions/dateTimeOperand"
        },
        "$dayOfMonth": {
          "$ref": "#/definitions/dateTimeOperand"
        },
        "$month": {
          "$ref": "#/definitions/dateTimeOperand"
        },
        "$year": {
          "$ref": "#/definitions/dateTimeOperand"
        }
      },
      "oneOf": [
        {
          "required": [
            "$numVal"
          ]
        },
        {
          "required": [
            "$numCast"
          ]
        },
        {
          "required": [
            "$dayOfWeek"
          ]
        },
```

```
        {
          "required": [
            "$dayOfMonth"
          ]
        },
        {
          "required": [
            "$month"
          ]
        },
        {
          "required": [
            "$year"
          ]
        }
      ],
      "additionalProperties": false
    },
    "hexOperand": {
      "type": "object",
      "properties": {
        "$hexVal": {
          "$ref": "#/definitions/hexLiteralPattern"
        },
        "$hexCast": {
          "$ref": "#/definitions/Value"
        }
      },
      "oneOf": [
        {
          "required": [
            "$hexVal"
          ]
        },
        {
          "required": [
            "$hexCast"
          ]
        }
      ],
      "additionalProperties": false
    },
    "boolOperand": {
      "type": "object",
      "properties": {
        "$boolean": {
          "type": "boolean"
        },
        "$boolCast": {
          "$ref": "#/definitions/Value"
        }
      },
```

```
      "oneOf": [
        {
          "required": [
            "$boolean"
          ]
        },
        {
          "required": [
            "$boolCast"
          ]
        }
      ],
      "additionalProperties": false
    },
    "timeOperand": {
      "type": "object",
      "properties": {
        "$timeVal": {
          "$ref": "#/definitions/timeLiteralPattern"
        },
        "$timeCast": {
          "anyOf": [
            {
              "$ref": "#/definitions/stringValue"
            },
            {
              "$ref": "#/definitions/dateTimeOperand"
            }
          ]
        }
      },
      "oneOf": [
        {
          "required": [
            "$timeVal"
          ]
        },
        {
          "required": [
            "$timeCast"
          ]
        }
      ],
      "additionalProperties": false
    },
    "Value": {
      "type": "object",
      "properties": {
        "$field": {
          "$ref": "#/definitions/FieldIdentifier"
        },
        "$strVal": {
```

```
      "$ref": "#/definitions/standardString"
    },
    "$attribute": {
      "$ref": "#/definitions/attributeItem"
    },
    "$numVal": {
      "type": "number"
    },
    "$hexVal": {
      "$ref": "#/definitions/hexLiteralPattern"
    },
    "$dateTimeVal": {
      "$ref": "#/definitions/dateTimeLiteralPattern"
    },
    "$timeVal": {
      "$ref": "#/definitions/timeLiteralPattern"
    },
    "$boolean": {
      "type": "boolean"
    },
    "$strCast": {
      "$ref": "#/definitions/Value"
    },
    "$numCast": {
      "$ref": "#/definitions/Value"
    },
    "$hexCast": {
      "$ref": "#/definitions/Value"
    },
    "$boolCast": {
      "$ref": "#/definitions/Value"
    },
    "$dateTimeCast": {
      "$ref": "#/definitions/stringValue"
    },
    "$timeCast": {
      "anyOf": [
        {
          "$ref": "#/definitions/stringValue"
        },
        {
          "$ref": "#/definitions/dateTimeOperand"
        }
      ]
    },
    "$dayOfWeek": {
      "$ref": "#/definitions/dateTimeOperand"
    },
    "$dayOfMonth": {
      "$ref": "#/definitions/dateTimeOperand"
    },
    "$month": {
```

```
        "$ref": "#/definitions/dateTimeOperand"
      },
      "$year": {
        "$ref": "#/definitions/dateTimeOperand"
      }
    },
    "oneOf": [
      {
        "required": [
          "$field"
        ]
      },
      {
        "required": [
          "$strVal"
        ]
      },
      {
        "required": [
          "$attribute"
        ]
      },
      {
        "required": [
          "$numVal"
        ]
      },
      {
        "required": [
          "$hexVal"
        ]
      },
      {
        "required": [
          "$dateTimeVal"
        ]
      },
      {
        "required": [
          "$timeVal"
        ]
      },
      {
        "required": [
          "$boolean"
        ]
      },
      {
        "required": [
          "$strCast"
        ]
      },
```

```
      {
        "required": [
          "$numCast"
        ]
      },
      {
        "required": [
          "$hexCast"
        ]
      },
      {
        "required": [
          "$boolCast"
        ]
      },
      {
        "required": [
          "$dateTimeCast"
        ]
      },
      {
        "required": [
          "$timeCast"
        ]
      },
      {
        "required": [
          "$dayOfWeek"
        ]
      },
      {
        "required": [
          "$dayOfMonth"
        ]
      },
      {
        "required": [
          "$month"
        ]
      },
      {
        "required": [
          "$year"
        ]
      }
    ],
    "additionalProperties": false
  },
  "stringValue": {
    "type": "object",
    "properties": {
      "$field": {
```

```
          "$ref": "#/definitions/FieldIdentifier"
        },
        "$strVal": {
          "$ref": "#/definitions/standardString"
        },
        "$strCast": {
          "$ref": "#/definitions/Value"
        },
        "$attribute": {
          "$ref": "#/definitions/attributeItem"
        }
      },
      "oneOf": [
        {
          "required": [
            "$field"
          ]
        },
        {
          "required": [
            "$strVal"
          ]
        },
        {
          "required": [
            "$strCast"
          ]
        },
        {
          "required": [
            "$attribute"
          ]
        }
      ],
      "additionalProperties": false
    },
    "stringNonFieldValue": {
      "type": "object",
      "properties": {
        "$strVal": {
          "$ref": "#/definitions/standardString"
        },
        "$strCast": {
          "$ref": "#/definitions/Value"
        },
        "$attribute": {
          "$ref": "#/definitions/attributeItem"
        }
      },
      "oneOf": [
        {
          "required": [
```

```
            "$strVal"
          ]
        },
        {
          "required": [
            "$strCast"
          ]
        },
        {
          "required": [
            "$attribute"
          ]
        }
      ],
      "additionalProperties": false
    },
    "comparisonItems": {
      "$ref": "#/definitions/equalityComparisonItems"
    },
    "stringItems": {
      "anyOf": [
        {
          "type": "array",
          "minItems": 2,
          "maxItems": 2,
          "items": [
            {
              "$ref": "#/definitions/stringValue"
            },
            {
              "$ref": "#/definitions/stringNonFieldValue"
            }
          ]
        },
        {
          "type": "array",
          "minItems": 2,
          "maxItems": 2,
          "items": [
            {
              "$ref": "#/definitions/stringNonFieldValue"
            },
            {
              "$ref": "#/definitions/stringValue"
            }
          ]
        }
      ]
    },
    "numericalComparisonItems": {
      "oneOf": [
        {
```

```
        "type": "array",
        "minItems": 2,
        "maxItems": 2,
        "items": {
          "$ref": "#/definitions/numericalOperand"
        }
      },
      {
        "type": "array",
        "minItems": 2,
        "maxItems": 2,
        "items": [
          {
            "$ref": "#/definitions/numericalOperand"
          },
          {
            "$ref": "#/definitions/fieldOperand"
          }
        ]
      },
      {
        "type": "array",
        "minItems": 2,
        "maxItems": 2,
        "items": [
          {
            "$ref": "#/definitions/fieldOperand"
          },
          {
            "$ref": "#/definitions/numericalOperand"
          }
        ]
      }
    ]
  },
  "hexComparisonItems": {
    "oneOf": [
      {
        "type": "array",
        "minItems": 2,
        "maxItems": 2,
        "items": {
          "$ref": "#/definitions/hexOperand"
        }
      },
      {
        "type": "array",
        "minItems": 2,
        "maxItems": 2,
        "items": [
          {
            "$ref": "#/definitions/hexOperand"
```

```
          },
          {
            "$ref": "#/definitions/fieldOperand"
          }
        ]
      },
      {
        "type": "array",
        "minItems": 2,
        "maxItems": 2,
        "items": [
          {
            "$ref": "#/definitions/fieldOperand"
          },
          {
            "$ref": "#/definitions/hexOperand"
          }
        ]
      }
    ]
  },
  "boolComparisonItems": {
    "oneOf": [
      {
        "type": "array",
        "minItems": 2,
        "maxItems": 2,
        "items": {
          "$ref": "#/definitions/boolOperand"
        }
      },
      {
        "type": "array",
        "minItems": 2,
        "maxItems": 2,
        "items": [
          {
            "$ref": "#/definitions/boolOperand"
          },
          {
            "$ref": "#/definitions/fieldOperand"
          }
        ]
      },
      {
        "type": "array",
        "minItems": 2,
        "maxItems": 2,
        "items": [
          {
            "$ref": "#/definitions/fieldOperand"
          },
```

```
          {
            "$ref": "#/definitions/boolOperand"
          }
        ]
      }
    ]
  },
  "dateTimeComparisonItems": {
    "oneOf": [
      {
        "type": "array",
        "minItems": 2,
        "maxItems": 2,
        "items": {
          "$ref": "#/definitions/dateTimeOperand"
        }
      },
      {
        "type": "array",
        "minItems": 2,
        "maxItems": 2,
        "items": [
          {
            "$ref": "#/definitions/dateTimeOperand"
          },
          {
            "$ref": "#/definitions/fieldOperand"
          }
        ]
      },
      {
        "type": "array",
        "minItems": 2,
        "maxItems": 2,
        "items": [
          {
            "$ref": "#/definitions/fieldOperand"
          },
          {
            "$ref": "#/definitions/dateTimeOperand"
          }
        ]
      }
    ]
  },
  "timeComparisonItems": {
    "oneOf": [
      {
        "type": "array",
        "minItems": 2,
        "maxItems": 2,
        "items": {
```

```
          "$ref": "#/definitions/timeOperand"
        }
      },
      {
        "type": "array",
        "minItems": 2,
        "maxItems": 2,
        "items": [
          {
            "$ref": "#/definitions/timeOperand"
          },
          {
            "$ref": "#/definitions/fieldOperand"
          }
        ]
      },
      {
        "type": "array",
        "minItems": 2,
        "maxItems": 2,
        "items": [
          {
            "$ref": "#/definitions/fieldOperand"
          },
          {
            "$ref": "#/definitions/timeOperand"
          }
        ]
      }
    ]
  },
  "orderedComparisonItems": {
    "oneOf": [
      {
        "$ref": "#/definitions/stringItems"
      },
      {
        "$ref": "#/definitions/numericalComparisonItems"
      },
      {
        "$ref": "#/definitions/hexComparisonItems"
      },
      {
        "$ref": "#/definitions/dateTimeComparisonItems"
      },
      {
        "$ref": "#/definitions/timeComparisonItems"
      }
    ]
  },
  "equalityComparisonItems": {
    "oneOf": [
```

```
        {
          "$ref": "#/definitions/orderedComparisonItems"
        },
        {
          "$ref": "#/definitions/boolComparisonItems"
        }
      ]
    },
    "matchExpression": {
      "type": "object",
      "properties": {
        "$match": {
          "type": "array",
          "minItems": 1,
          "items": {
            "$ref": "#/definitions/matchExpression"
          }
        },
        "$eq": {
          "$ref": "#/definitions/equalityComparisonItems"
        },
        "$ne": {
          "$ref": "#/definitions/equalityComparisonItems"
        },
        "$gt": {
          "$ref": "#/definitions/orderedComparisonItems"
        },
        "$ge": {
          "$ref": "#/definitions/orderedComparisonItems"
        },
        "$lt": {
          "$ref": "#/definitions/orderedComparisonItems"
        },
        "$le": {
          "$ref": "#/definitions/orderedComparisonItems"
        },
        "$contains": {
          "$ref": "#/definitions/stringItems"
        },
        "$starts-with": {
          "$ref": "#/definitions/stringItems"
        },
        "$ends-with": {
          "$ref": "#/definitions/stringItems"
        },
        "$regex": {
          "$ref": "#/definitions/stringItems"
        }
      },
      "oneOf": [
        {
          "required": [
```

```
          "$eq"
        ]
      },
      {
        "required": [
          "$ne"
        ]
      },
      {
        "required": [
          "$gt"
        ]
      },
      {
        "required": [
          "$ge"
        ]
      },
      {
        "required": [
          "$lt"
        ]
      },
      {
        "required": [
          "$le"
        ]
      },
      {
        "required": [
          "$contains"
        ]
      },
      {
        "required": [
          "$starts-with"
        ]
      },
      {
        "required": [
          "$ends-with"
        ]
      },
      {
        "required": [
          "$regex"
        ]
      },
      {
        "required": [
          "$match"
        ]
```

```
        }
      ],
      "additionalProperties": false
    },
    "logicalExpression": {
      "type": "object",
      "properties": {
        "$and": {
          "type": "array",
          "minItems": 2,
          "items": {
            "$ref": "#/definitions/logicalExpression"
          }
        },
        "$match": {
          "type": "array",
          "minItems": 1,
          "items": {
            "$ref": "#/definitions/matchExpression"
          }
        },
        "$or": {
          "type": "array",
          "minItems": 2,
          "items": {
            "$ref": "#/definitions/logicalExpression"
          }
        },
        "$not": {
          "$ref": "#/definitions/logicalExpression"
        },
        "$eq": {
          "$ref": "#/definitions/equalityComparisonItems"
        },
        "$ne": {
          "$ref": "#/definitions/equalityComparisonItems"
        },
        "$gt": {
          "$ref": "#/definitions/orderedComparisonItems"
        },
        "$ge": {
          "$ref": "#/definitions/orderedComparisonItems"
        },
        "$lt": {
          "$ref": "#/definitions/orderedComparisonItems"
        },
        "$le": {
          "$ref": "#/definitions/orderedComparisonItems"
        },
        "$contains": {
          "$ref": "#/definitions/stringItems"
        },
```

```
      "$starts-with": {
        "$ref": "#/definitions/stringItems"
      },
      "$ends-with": {
        "$ref": "#/definitions/stringItems"
      },
      "$regex": {
        "$ref": "#/definitions/stringItems"
      },
      "$boolean": {
        "type": "boolean"
      },
      "$boolCast": {
        "$ref": "#/definitions/Value"
      }
    },
    "oneOf": [
      {
        "required": [
          "$and"
        ]
      },
      {
        "required": [
          "$or"
        ]
      },
      {
        "required": [
          "$not"
        ]
      },
      {
        "required": [
          "$eq"
        ]
      },
      {
        "required": [
          "$ne"
        ]
      },
      {
        "required": [
          "$gt"
        ]
      },
      {
        "required": [
          "$ge"
        ]
      },
```

```
      {
        "required": [
          "$lt"
        ]
      },
      {
        "required": [
          "$le"
        ]
      },
      {
        "required": [
          "$contains"
        ]
      },
      {
        "required": [
          "$starts-with"
        ]
      },
      {
        "required": [
          "$ends-with"
        ]
      },
      {
        "required": [
          "$regex"
        ]
      },
      {
        "required": [
          "$boolean"
        ]
      },
      {
        "required": [
          "$boolCast"
        ]
      },
      {
        "required": [
          "$match"
        ]
      }
    ],
    "additionalProperties": false
  },
  "attributeItem": {
    "oneOf": [
      {
        "required": [
```

```
          "CLAIM"
        ]
      },
      {
        "required": [
          "GLOBAL"
        ]
      },
      {
        "required": [
          "REFERENCE"
        ]
      }
    ],
    "properties": {
      "CLAIM": {
        "type": "string"
      },
      "GLOBAL": {
        "type": "string",
        "enum": [
          "LOCALNOW",
          "UTCNOW",
          "CLIENTNOW",
          "ANONYMOUS"
        ]
      },
      "REFERENCE": {
        "$ref": "#/definitions/ReferenceIdentifier"
      }
    },
    "additionalProperties": false
  },
  "objectItem": {
    "oneOf": [
      {
        "required": [
          "ROUTE"
        ]
      },
      {
        "required": [
          "IDENTIFIABLE"
        ]
      },
      {
        "required": [
          "REFERABLE"
        ]
      },
      {
        "required": [
```

```
          "FRAGMENT"
        ]
      },
      {
        "required": [
          "DESCRIPTOR"
        ]
      }
    ],
    "properties": {
      "ROUTE": {
        "type": "string"
      },
      "IDENTIFIABLE": {
        "$ref": "#/definitions/IdentifiableIdentifier"
      },
      "REFERABLE": {
        "$ref": "#/definitions/ReferableIdentifier"
      },
      "FRAGMENT": {
        "$ref": "#/definitions/FragmentIdentifier"
      },
      "DESCRIPTOR": {
        "$ref": "#/definitions/DescriptorIdentifier"
      }
    },
    "additionalProperties": false
  },
  "rightsEnum": {
    "type": "string",
    "enum": [
      "CREATE",
      "READ",
      "UPDATE",
      "DELETE",
      "EXECUTE",
      "VIEW",
      "ALL"
    ],
    "additionalProperties": false
  },
  "ACL": {
    "type": "object",
    "properties": {
      "ATTRIBUTES": {
        "type": "array",
        "items": {
          "$ref": "#/definitions/attributeItem"
        }
      },
      "USEATTRIBUTES": {
        "type": "string"
```

```
      },
      "RIGHTS": {
        "type": "array",
        "items": {
          "$ref": "#/definitions/rightsEnum"
        }
      },
      "ACCESS": {
        "type": "string",
        "enum": [
          "ALLOW",
          "DISABLED"
        ]
      }
    },
    "required": [
      "RIGHTS",
      "ACCESS"
    ],
    "anyOf": [
      {
        "required": [
          "ATTRIBUTES"
        ]
      },
      {
        "required": [
          "USEATTRIBUTES"
        ]
      }
    ],
    "additionalProperties": false
  },
  "AccessPermissionRule": {
    "type": "object",
    "properties": {
      "ACL": {
        "$ref": "#/definitions/ACL"
      },
      "USEACL": {
        "type": "string"
      },
      "OBJECTS": {
        "type": "array",
        "items": {
          "$ref": "#/definitions/objectItem"
        },
        "additionalProperties": false
      },
      "USEOBJECTS": {
        "type": "array",
        "items": {
```

```
          "type": "string"
        }
      },
      "FORMULA": {
        "$ref": "#/definitions/logicalExpression",
        "additionalProperties": false
      },
      "USEFORMULA": {
        "type": "string"
      },
      "FILTER": {
        "$ref": "#/definitions/SecurityQueryFilter",
        "additionalProperties": false
      },
      "FILTERLIST": {
        "type": "array",
        "items": {
          "$ref": "#/definitions/SecurityQueryFilter"
        }
      }
    },
    "allOf": [
      {
        "oneOf": [
          {
            "required": [
              "ACL"
            ]
          },
          {
            "required": [
              "USEACL"
            ]
          }
        ]
      },
      {
        "anyOf": [
          {
            "required": [
              "OBJECTS"
            ]
          },
          {
            "required": [
              "USEOBJECTS"
            ]
          }
        ]
      },
      {
        "oneOf": [
```

```
          {
            "required": [
              "FORMULA"
            ]
          },
          {
            "required": [
              "USEFORMULA"
            ]
          }
        ]
      }
    ],
    "additionalProperties": false
  },
  "SecurityQueryFilter": {
    "type": "object",
    "properties": {
      "FRAGMENT": {
        "$ref": "#/definitions/FragmentFieldIdentifier"
      },
      "CONDITION": {
        "$ref": "#/definitions/logicalExpression"
      },
      "USEFORMULA": {
        "type": "string"
      }
    },
    "required": [
      "FRAGMENT"
    ],
    "oneOf": [
      {
        "required": [
          "CONDITION"
        ]
      },
      {
        "required": [
          "USEFORMULA"
        ]
      }
    ],
    "additionalProperties": false
  },
  "QueryFilter": {
    "type": "object",
    "properties": {
      "$fragment": {
        "$ref": "#/definitions/FragmentFieldIdentifier"
      },
      "$condition": {
```

```
          "$ref": "#/definitions/logicalExpression"
        }
      },
      "required": [
        "$fragment",
        "$condition"
      ],
      "additionalProperties": false
    },
    "Query": {
      "type": "object",
      "properties": {
        "$select": {
          "type": "string",
          "pattern": "^id$"
        },
        "$condition": {
          "$ref": "#/definitions/logicalExpression"
        },
        "$filters": {
          "type": "array",
          "items": {
            "$ref": "#/definitions/QueryFilter"
          }
        }
      },
      "required": [
        "$condition"
      ],
      "additionalProperties": false
    },
    "AllAccessPermissionRules": {
      "type": "object",
      "properties": {
        "DEFATTRIBUTES": {
          "type": "array",
          "items": {
            "type": "object",
            "properties": {
              "name": {
                "type": "string"
              },
              "attributes": {
                "type": "array",
                "items": {
                  "$ref": "#/definitions/attributeItem"
                }
              },
              "USEATTRIBUTES": {
                "type": "array",
                "items": {
                  "type": "string"
```

```
            }
          }
        },
        "required": [
          "name"
        ],
        "oneOf": [
          {
            "required": [
              "attributes"
            ]
          },
          {
            "required": [
              "USEATTRIBUTES"
            ]
          }
        ],
        "additionalProperties": false
      }
    },
    "DEFACLS": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "name": {
            "type": "string"
          },
          "acl": {
            "$ref": "#/definitions/ACL"
          }
        },
        "required": [
          "name",
          "acl"
        ],
        "additionalProperties": false
      }
    },
    "DEFOBJECTS": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "name": {
            "type": "string"
          },
          "objects": {
            "type": "array",
            "items": {
              "$ref": "#/definitions/objectItem"
```

```
            }
          },
          "USEOBJECTS": {
            "type": "array",
            "items": {
              "type": "string"
            }
          }
        },
        "required": [
          "name"
        ],
        "oneOf": [
          {
            "required": [
              "objects"
            ]
          },
          {
            "required": [
              "USEOBJECTS"
            ]
          }
        ],
        "additionalProperties": false
      }
    },
    "DEFFORMULAS": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "name": {
            "type": "string"
          },
          "formula": {
            "$ref": "#/definitions/logicalExpression"
          }
        },
        "required": [
          "name",
          "formula"
        ],
        "additionalProperties": false
      }
    },
    "rules": {
      "type": "array",
      "items": {
        "$ref": "#/definitions/AccessPermissionRule"
      }
    }
```

```
      },
      "required": [
        "rules"
      ],
      "additionalProperties": false
    }
  },
  "additionalProperties": false
}
```

Explanation of the Access Rule Model JSON schema

General

The AAS Access Rule Model can be used to describe access rules. Whether and how access rules are enforced is beyond the specification of the model for access control. The parties involved are supposed to agree on governance and policies.

Access Rule Model

The Access Rules BNF grammar is expressed in JSON schema syntax. Please refer to the detailed step by step explanation of the grammar.

```
{
  "definitions": {
    "attributeItem": {
      "oneOf": [
        {
          "required": [
            "CLAIM"
          ]
        },
        {
          "required": [
            "GLOBAL"
          ]
        },
        {
          "required": [
            "REFERENCE"
          ]
        }
      ],
      "properties": {
        "CLAIM": {
          "type": "string"
        },
        "GLOBAL": {
          "type": "string",
          "enum": [
            "LOCALNOW",
```

```
            "UTCNOW",
            "CLIENTNOW",
            "ANONYMOUS"
          ]
        },
        "REFERENCE": {
          "$ref": "aas-queries-and-access-rules-
schema.json#/definitions/ReferenceIdentifier"
        }
      },
      "additionalProperties": false
    },
    "objectItem": {
      "oneOf": [
        {
          "required": [
            "ROUTE"
          ]
        },
        {
          "required": [
            "IDENTIFIABLE"
          ]
        },
        {
          "required": [
            "REFERABLE"
          ]
        },
        {
          "required": [
            "FRAGMENT"
          ]
        },
        {
          "required": [
            "DESCRIPTOR"
          ]
        }
      ],
      "properties": {
        "ROUTE": {
          "type": "string"
        },
        "IDENTIFIABLE": {
          "$ref": "aas-queries-and-access-rules-
schema.json#/definitions/IdentifiableIdentifier"
        },
        "REFERABLE": {
          "$ref": "aas-queries-and-access-rules-
schema.json#/definitions/ReferableIdentifier"
        },
```

```
        "FRAGMENT": {
          "$ref": "aas-queries-and-access-rules-
schema.json#/definitions/FragmentIdentifier"
        },
        "DESCRIPTOR": {
          "$ref": "aas-queries-and-access-rules-
schema.json#/definitions/DescriptorIdentifier"
        }
      },
      "additionalProperties": false
    },
    "rightsEnum": {
      "type": "string",
      "enum": [
        "CREATE",
        "READ",
        "UPDATE",
        "DELETE",
        "EXECUTE",
        "VIEW",
        "ALL"
      ],
      "additionalProperties": false
    },
    "ACL": {
      "type": "object",
      "properties": {
        "ATTRIBUTES": {
          "type": "array",
          "items": {
            "$ref": "#/definitions/attributeItem"
          }
        },
        "USEATTRIBUTES": {
          "type": "string"
        },
        "RIGHTS": {
          "type": "array",
          "items": {
            "$ref": "#/definitions/rightsEnum"
          }
        },
        "ACCESS": {
          "type": "string",
          "enum": [
            "ALLOW",
            "DISABLED"
          ]
        }
      },
      "required": [
        "RIGHTS",
```

```
        "ACCESS"
      ],
      "anyOf": [
        {
          "required": [
            "ATTRIBUTES"
          ]
        },
        {
          "required": [
            "USEATTRIBUTES"
          ]
        }
      ],
      "additionalProperties": false
    },
    "AccessPermissionRule": {
      "type": "object",
      "properties": {
        "ACL": {
          "$ref": "#/definitions/ACL"
        },
        "USEACL": {
          "type": "string"
        },
        "OBJECTS": {
          "type": "array",
          "items": {
            "$ref": "#/definitions/objectItem"
          },
          "additionalProperties": false
        },
        "USEOBJECTS": {
          "type": "array",
          "items": {
            "type": "string"
          }
        },
        "FORMULA": {
          "$ref": "aas-queries-and-access-rules-
schema.json#/definitions/logicalExpression",
          "additionalProperties": false
        },
        "USEFORMULA": {
          "type": "string"
        },
        "FILTER": {
          "$ref": "#/definitions/SecurityQueryFilter",
          "additionalProperties": false
        },
        "FILTERLIST": {
          "type": "array",
```

```
        "items": {
          "$ref": "#/definitions/SecurityQueryFilter"
        }
      }
    },
    "allOf": [
      {
        "oneOf": [
          {
            "required": [
              "ACL"
            ]
          },
          {
            "required": [
              "USEACL"
            ]
          }
        ]
      },
      {
        "anyOf": [
          {
            "required": [
              "OBJECTS"
            ]
          },
          {
            "required": [
              "USEOBJECTS"
            ]
          }
        ]
      },
      {
        "oneOf": [
          {
            "required": [
              "FORMULA"
            ]
          },
          {
            "required": [
              "USEFORMULA"
            ]
          }
        ]
      }
    ],
    "additionalProperties": false
  },
  "SecurityQueryFilter": {
```

```
      "type": "object",
      "properties": {
        "FRAGMENT": {
          "$ref": "aas-queries-and-access-rules-
schema.json#/definitions/FragmentFieldIdentifier"
        },
        "CONDITION": {
          "$ref": "aas-queries-and-access-rules-schema.json#/definitions/logicalExpression"
        },
        "USEFORMULA": {
          "type": "string"
        }
      },
      "required": [
        "FRAGMENT"
      ],
      "oneOf": [
        {
          "required": [
            "CONDITION"
          ]
        },
        {
          "required": [
            "USEFORMULA"
          ]
        }
      ],
      "additionalProperties": false
    }
  },
  "type": "object",
  "properties": {
    "AllAccessPermissionRules": {
      "type": "object",
      "properties": {
        "DEFATTRIBUTES": {
          "type": "array",
          "items": {
            "type": "object",
            "properties": {
              "name": {
                "type": "string"
              },
              "attributes": {
                "type": "array",
                "items": {
                  "$ref": "#/definitions/attributeItem"
                }
              },
              "USEATTRIBUTES": {
                "type": "array",
```

```
            "items": {
              "type": "string"
            }
          }
        },
        "required": [
          "name"
        ],
        "oneOf": [
          {
            "required": [
              "attributes"
            ]
          },
          {
            "required": [
              "USEATTRIBUTES"
            ]
          }
        ],
        "additionalProperties": false
      }
    },
    "DEFACLS": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "name": {
            "type": "string"
          },
          "acl": {
            "$ref": "#/definitions/ACL"
          }
        },
        "required": [
          "name",
          "acl"
        ],
        "additionalProperties": false
      }
    },
    "DEFOBJECTS": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "name": {
            "type": "string"
          },
          "objects": {
            "type": "array",
```

```
              "items": {
                "$ref": "#/definitions/objectItem"
              }
            },
            "USEOBJECTS": {
              "type": "array",
              "items": {
                "type": "string"
              }
            }
          },
          "required": [
            "name"
          ],
          "oneOf": [
            {
              "required": [
                "objects"
              ]
            },
            {
              "required": [
                "USEOBJECTS"
              ]
            }
          ],
          "additionalProperties": false
        }
      },
      "DEFFORMULAS": {
        "type": "array",
        "items": {
          "type": "object",
          "properties": {
            "name": {
              "type": "string"
            },
            "formula": {
              "$ref": "aas-queries-and-access-rules-
schema.json#/definitions/logicalExpression"
            }
          },
          "required": [
            "name",
            "formula"
          ],
          "additionalProperties": false
        }
      },
      "rules": {
        "type": "array",
        "items": {
```

```
            "$ref": "#/definitions/AccessPermissionRule"
          }
        }
      },
      "required": [
        "rules"
      ],
      "additionalProperties": false
    }
  },
  "required": [
    "AllAccessPermissionRules"
  ],
  "additionalProperties": false
}
```

Formulas and logical expressions

A detailed step by step explanation follows below the JSON schema code.

```
{
  "definitions": {
    "standardString": {
      "type": "string",
      "pattern": "^[A-Za-z0-9/\\*\\[\\]\\(\\) _@#\\\\+\\-\\.,:\\$\\^]+$"
    },
    "FieldIdentifier": {
      "type": "string",
      "pattern":
"^(?:\\$aas#(?:idShort|id|assetInformation\\.assetKind|assetInformation\\.assetType|assetIn
formation\\.globalAssetId|assetInformation\\.specificAssetIds\\[(?:0|[1-9][0-
9]*)?\\]\\.(?:name|value|externalSubjectId(?:\\.(?:type|keys\\[(?:0|[1-9][0-
9]*)?\\]\\.(?:type|value)))?)|submodels\\[(?:0|[1-9][0-
9]*)?\\](?:\\.(?:type|keys\\[(?:0|[1-9][0-
9]*)?\\]\\.(?:type|value)))?)|\\$sm#(?:semanticId(?:\\.(?:type|keys\\[(?:0|[1-9][0-
9]*)?\\]\\.(?:type|value)))?|supplementalSemanticIds(?:\\[(?:0|[1-9][0-
9]*)?\\])?(?:\\.(?:type|keys\\[(?:0|[1-9][0-
9]*)?\\]\\.(?:type|value)))?|idShort|id)|\\$sme(?:\\.[A-Za-z](?:[A-Za-z0-9_-]*[A-Za-z0-
9_])?(?:\\[(?:0|[1-9][0-9]*)?\\])*(?:\\.[A-Za-z](?:[A-Za-z0-9_-]*[A-Za-z0-
9_])?(?:\\[(?:0|[1-9][0-9]*)?\\])*)*)?#(?:semanticId(?:\\.(?:type|keys\\[(?:0|[1-9][0-
9]*)?\\]\\.(?:type|value)))?|supplementalSemanticIds(?:\\[(?:0|[1-9][0-
9]*)?\\])?(?:\\.(?:type|keys\\[(?:0|[1-9][0-
9]*)?\\]\\.(?:type|value)))?|idShort|value|valueType|language)|\\$cd#(?:idShort|id)|\\$aasd
esc#(?:idShort|id|assetKind|assetType|globalAssetId|specificAssetIds\\[(?:0|[1-9][0-
9]*)?\\]\\.(?:name|value|externalSubjectId(?:\\.(?:type|keys\\[(?:0|[1-9][0-
9]*)?\\]\\.(?:type|value)))?)|endpoints\\[(?:0|[1-9][0-
9]*)?\\]\\.(?:interface|protocolinformation\\.href)|submodelDescriptors\\[(?:0|[1-9][0-
9]*)?\\]\\.(?:semanticId(?:\\.(?:type|keys\\[(?:0|[1-9][0-
9]*)?\\]\\.(?:type|value)))?|supplementalSemanticIds(?:\\[(?:0|[1-9][0-
9]*)?\\])?(?:\\.(?:type|keys\\[(?:0|[1-9][0-
9]*)?\\]\\.(?:type|value)))?|idShort|id|endpoints\\[(?:0|[1-9][0-
```

```
9]*)?\\]\\.(?:interface|protocolinformation\\.href)))|\\$smdesc#(?:semanticId(?:\\.(?:type|
keys\\[(?:0|[1-9][0-9]*)?\\]\\.(?:type|value)))?|supplementalSemanticIds(?:\\[(?:0|[1-9][0-
9]*)?\\])?(?:\\.(?:type|keys\\[(?:0|[1-9][0-
9]*)?\\]\\.(?:type|value)))?|idShort|id|endpoints\\[(?:0|[1-9][0-
9]*)?\\]\\.(?:interface|protocolinformation\\.href)))$"
    },
    "hexLiteralPattern": {
      "type": "string",
      "pattern": "^16#[0-9A-F]+$"
    },
    "dateTimeLiteralPattern": {
      "type": "string",
      "pattern": "^-?(([1-9][0-9][0-9][0-9]+)|(0[0-9][0-9][0-9]))-((0[1-9])|(1[0-2]))-
((0[1-9])|([12][0-9])|(3[01]))T(((([01][0-9])|(2[0-3])):[0-5][0-9]:([0-5][0-9])(\\.[0-
9]+)?)|24:00:00(\\.0+)?)(Z|(\\+|-)((0[0-9]|1[0-3]):[0-5][0-9]|14:00))?$"
    },
    "timeLiteralPattern": {
      "type": "string",
      "pattern": "^(([01][0-9]|2[0-3]):[0-5][0-9]:[0-5][0-9](\\.[0-
9]+)?|(24:00:00(\\.0+)?))(Z|(\\+|-)((0[0-9]|1[0-3]):[0-5][0-9]|14:00))?$"
    },
    "dateTimeOperand": {
      "type": "object",
      "properties": {
        "$dateTimeVal": {
          "$ref": "#/definitions/dateTimeLiteralPattern"
        },
        "$dateTimeCast": {
          "$ref": "#/definitions/stringValue"
        },
        "$attribute": {
          "$ref": "aas-queries-and-access-rules-
schema.json#/definitions/dateTimeAttributeItem"
        }
      },
      "oneOf": [
        {
          "required": [
            "$dateTimeVal"
          ]
        },
        {
          "required": [
            "$dateTimeCast"
          ]
        },
        {
          "required": [
            "$attribute"
          ]
        }
      ],
```

```
      "additionalProperties": false
    },
    "fieldOperand": {
      "type": "object",
      "properties": {
        "$field": {
          "$ref": "#/definitions/FieldIdentifier"
        }
      },
      "required": [
        "$field"
      ],
      "additionalProperties": false
    },
    "numericalOperand": {
      "type": "object",
      "properties": {
        "$numVal": {
          "type": "number"
        },
        "$numCast": {
          "$ref": "#/definitions/Value"
        },
        "$dayOfWeek": {
          "$ref": "#/definitions/dateTimeOperand"
        },
        "$dayOfMonth": {
          "$ref": "#/definitions/dateTimeOperand"
        },
        "$month": {
          "$ref": "#/definitions/dateTimeOperand"
        },
        "$year": {
          "$ref": "#/definitions/dateTimeOperand"
        }
      },
      "oneOf": [
        {
          "required": [
            "$numVal"
          ]
        },
        {
          "required": [
            "$numCast"
          ]
        },
        {
          "required": [
            "$dayOfWeek"
          ]
        },
```

```
        {
          "required": [
            "$dayOfMonth"
          ]
        },
        {
          "required": [
            "$month"
          ]
        },
        {
          "required": [
            "$year"
          ]
        }
      ],
      "additionalProperties": false
    },
    "hexOperand": {
      "type": "object",
      "properties": {
        "$hexVal": {
          "$ref": "#/definitions/hexLiteralPattern"
        },
        "$hexCast": {
          "$ref": "#/definitions/Value"
        }
      },
      "oneOf": [
        {
          "required": [
            "$hexVal"
          ]
        },
        {
          "required": [
            "$hexCast"
          ]
        }
      ],
      "additionalProperties": false
    },
    "boolOperand": {
      "type": "object",
      "properties": {
        "$boolean": {
          "type": "boolean"
        },
        "$boolCast": {
          "$ref": "#/definitions/Value"
        }
      },
```

```
      "oneOf": [
        {
          "required": [
            "$boolean"
          ]
        },
        {
          "required": [
            "$boolCast"
          ]
        }
      ],
      "additionalProperties": false
    },
    "timeOperand": {
      "type": "object",
      "properties": {
        "$timeVal": {
          "$ref": "#/definitions/timeLiteralPattern"
        },
        "$timeCast": {
          "anyOf": [
            {
              "$ref": "#/definitions/stringValue"
            },
            {
              "$ref": "#/definitions/dateTimeOperand"
            }
          ]
        }
      },
      "oneOf": [
        {
          "required": [
            "$timeVal"
          ]
        },
        {
          "required": [
            "$timeCast"
          ]
        }
      ],
      "additionalProperties": false
    },
    "Value": {
      "type": "object",
      "properties": {
        "$field": {
          "$ref": "#/definitions/FieldIdentifier"
        },
        "$strVal": {
```

```
      "$ref": "#/definitions/standardString"
    },
    "$attribute": {
      "$ref": "aas-queries-and-access-rules-schema.json#/definitions/attributeItem"
    },
    "$numVal": {
      "type": "number"
    },
    "$hexVal": {
      "$ref": "#/definitions/hexLiteralPattern"
    },
    "$dateTimeVal": {
      "$ref": "#/definitions/dateTimeLiteralPattern"
    },
    "$timeVal": {
      "$ref": "#/definitions/timeLiteralPattern"
    },
    "$boolean": {
      "type": "boolean"
    },
    "$strCast": {
      "$ref": "#/definitions/Value"
    },
    "$numCast": {
      "$ref": "#/definitions/Value"
    },
    "$hexCast": {
      "$ref": "#/definitions/Value"
    },
    "$boolCast": {
      "$ref": "#/definitions/Value"
    },
    "$dateTimeCast": {
      "$ref": "#/definitions/stringValue"
    },
    "$timeCast": {
      "anyOf": [
        {
          "$ref": "#/definitions/stringValue"
        },
        {
          "$ref": "#/definitions/dateTimeOperand"
        }
      ]
    },
    "$dayOfWeek": {
      "$ref": "#/definitions/dateTimeOperand"
    },
    "$dayOfMonth": {
      "$ref": "#/definitions/dateTimeOperand"
    },
    "$month": {
```

```
        "$ref": "#/definitions/dateTimeOperand"
      },
      "$year": {
        "$ref": "#/definitions/dateTimeOperand"
      }
    },
    "oneOf": [
      {
        "required": [
          "$field"
        ]
      },
      {
        "required": [
          "$strVal"
        ]
      },
      {
        "required": [
          "$attribute"
        ]
      },
      {
        "required": [
          "$numVal"
        ]
      },
      {
        "required": [
          "$hexVal"
        ]
      },
      {
        "required": [
          "$dateTimeVal"
        ]
      },
      {
        "required": [
          "$timeVal"
        ]
      },
      {
        "required": [
          "$boolean"
        ]
      },
      {
        "required": [
          "$strCast"
        ]
      },
```

```
      {
        "required": [
          "$numCast"
        ]
      },
      {
        "required": [
          "$hexCast"
        ]
      },
      {
        "required": [
          "$boolCast"
        ]
      },
      {
        "required": [
          "$dateTimeCast"
        ]
      },
      {
        "required": [
          "$timeCast"
        ]
      },
      {
        "required": [
          "$dayOfWeek"
        ]
      },
      {
        "required": [
          "$dayOfMonth"
        ]
      },
      {
        "required": [
          "$month"
        ]
      },
      {
        "required": [
          "$year"
        ]
      }
    ],
    "additionalProperties": false
  },
  "stringValue": {
    "type": "object",
    "properties": {
      "$field": {
```

```
          "$ref": "#/definitions/FieldIdentifier"
        },
        "$strVal": {
          "$ref": "#/definitions/standardString"
        },
        "$strCast": {
          "$ref": "#/definitions/Value"
        },
        "$attribute": {
          "$ref": "aas-queries-and-access-rules-schema.json#/definitions/attributeItem"
        }
      },
      "oneOf": [
        {
          "required": [
            "$field"
          ]
        },
        {
          "required": [
            "$strVal"
          ]
        },
        {
          "required": [
            "$strCast"
          ]
        },
        {
          "required": [
            "$attribute"
          ]
        }
      ],
      "additionalProperties": false
    },
    "stringNonFieldValue": {
      "type": "object",
      "properties": {
        "$strVal": {
          "$ref": "#/definitions/standardString"
        },
        "$strCast": {
          "$ref": "#/definitions/Value"
        },
        "$attribute": {
          "$ref": "aas-queries-and-access-rules-schema.json#/definitions/attributeItem"
        }
      },
      "oneOf": [
        {
          "required": [
```

```
            "$strVal"
          ]
        },
        {
          "required": [
            "$strCast"
          ]
        },
        {
          "required": [
            "$attribute"
          ]
        }
      ],
      "additionalProperties": false
    },
    "comparisonItems": {
      "$ref": "#/definitions/equalityComparisonItems"
    },
    "stringItems": {
      "anyOf": [
        {
          "type": "array",
          "minItems": 2,
          "maxItems": 2,
          "items": [
            {
              "$ref": "#/definitions/stringValue"
            },
            {
              "$ref": "#/definitions/stringNonFieldValue"
            }
          ]
        },
        {
          "type": "array",
          "minItems": 2,
          "maxItems": 2,
          "items": [
            {
              "$ref": "#/definitions/stringNonFieldValue"
            },
            {
              "$ref": "#/definitions/stringValue"
            }
          ]
        }
      ]
    },
    "numericalComparisonItems": {
      "oneOf": [
        {
```

```
        "type": "array",
        "minItems": 2,
        "maxItems": 2,
        "items": {
          "$ref": "#/definitions/numericalOperand"
        }
      },
      {
        "type": "array",
        "minItems": 2,
        "maxItems": 2,
        "items": [
          {
            "$ref": "#/definitions/numericalOperand"
          },
          {
            "$ref": "#/definitions/fieldOperand"
          }
        ]
      },
      {
        "type": "array",
        "minItems": 2,
        "maxItems": 2,
        "items": [
          {
            "$ref": "#/definitions/fieldOperand"
          },
          {
            "$ref": "#/definitions/numericalOperand"
          }
        ]
      }
    ]
  },
  "hexComparisonItems": {
    "oneOf": [
      {
        "type": "array",
        "minItems": 2,
        "maxItems": 2,
        "items": {
          "$ref": "#/definitions/hexOperand"
        }
      },
      {
        "type": "array",
        "minItems": 2,
        "maxItems": 2,
        "items": [
          {
            "$ref": "#/definitions/hexOperand"
```

```
          },
          {
            "$ref": "#/definitions/fieldOperand"
          }
        ]
      },
      {
        "type": "array",
        "minItems": 2,
        "maxItems": 2,
        "items": [
          {
            "$ref": "#/definitions/fieldOperand"
          },
          {
            "$ref": "#/definitions/hexOperand"
          }
        ]
      }
    ]
  },
  "boolComparisonItems": {
    "oneOf": [
      {
        "type": "array",
        "minItems": 2,
        "maxItems": 2,
        "items": {
          "$ref": "#/definitions/boolOperand"
        }
      },
      {
        "type": "array",
        "minItems": 2,
        "maxItems": 2,
        "items": [
          {
            "$ref": "#/definitions/boolOperand"
          },
          {
            "$ref": "#/definitions/fieldOperand"
          }
        ]
      },
      {
        "type": "array",
        "minItems": 2,
        "maxItems": 2,
        "items": [
          {
            "$ref": "#/definitions/fieldOperand"
          },
```

```
          {
            "$ref": "#/definitions/boolOperand"
          }
        ]
      }
    ]
  },
  "dateTimeComparisonItems": {
    "oneOf": [
      {
        "type": "array",
        "minItems": 2,
        "maxItems": 2,
        "items": {
          "$ref": "#/definitions/dateTimeOperand"
        }
      },
      {
        "type": "array",
        "minItems": 2,
        "maxItems": 2,
        "items": [
          {
            "$ref": "#/definitions/dateTimeOperand"
          },
          {
            "$ref": "#/definitions/fieldOperand"
          }
        ]
      },
      {
        "type": "array",
        "minItems": 2,
        "maxItems": 2,
        "items": [
          {
            "$ref": "#/definitions/fieldOperand"
          },
          {
            "$ref": "#/definitions/dateTimeOperand"
          }
        ]
      }
    ]
  },
  "timeComparisonItems": {
    "oneOf": [
      {
        "type": "array",
        "minItems": 2,
        "maxItems": 2,
        "items": {
```

```
          "$ref": "#/definitions/timeOperand"
        }
      },
      {
        "type": "array",
        "minItems": 2,
        "maxItems": 2,
        "items": [
          {
            "$ref": "#/definitions/timeOperand"
          },
          {
            "$ref": "#/definitions/fieldOperand"
          }
        ]
      },
      {
        "type": "array",
        "minItems": 2,
        "maxItems": 2,
        "items": [
          {
            "$ref": "#/definitions/fieldOperand"
          },
          {
            "$ref": "#/definitions/timeOperand"
          }
        ]
      }
    ]
  },
  "orderedComparisonItems": {
    "oneOf": [
      {
        "$ref": "#/definitions/stringItems"
      },
      {
        "$ref": "#/definitions/numericalComparisonItems"
      },
      {
        "$ref": "#/definitions/hexComparisonItems"
      },
      {
        "$ref": "#/definitions/dateTimeComparisonItems"
      },
      {
        "$ref": "#/definitions/timeComparisonItems"
      }
    ]
  },
  "equalityComparisonItems": {
    "oneOf": [
```

```
        {
          "$ref": "#/definitions/orderedComparisonItems"
        },
        {
          "$ref": "#/definitions/boolComparisonItems"
        }
      ]
    },
    "matchExpression": {
      "type": "object",
      "properties": {
        "$match": {
          "type": "array",
          "minItems": 1,
          "items": {
            "$ref": "#/definitions/matchExpression"
          }
        },
        "$eq": {
          "$ref": "#/definitions/equalityComparisonItems"
        },
        "$ne": {
          "$ref": "#/definitions/equalityComparisonItems"
        },
        "$gt": {
          "$ref": "#/definitions/orderedComparisonItems"
        },
        "$ge": {
          "$ref": "#/definitions/orderedComparisonItems"
        },
        "$lt": {
          "$ref": "#/definitions/orderedComparisonItems"
        },
        "$le": {
          "$ref": "#/definitions/orderedComparisonItems"
        },
        "$contains": {
          "$ref": "#/definitions/stringItems"
        },
        "$starts-with": {
          "$ref": "#/definitions/stringItems"
        },
        "$ends-with": {
          "$ref": "#/definitions/stringItems"
        },
        "$regex": {
          "$ref": "#/definitions/stringItems"
        }
      },
      "oneOf": [
        {
          "required": [
```

```
          "$eq"
        ]
      },
      {
        "required": [
          "$ne"
        ]
      },
      {
        "required": [
          "$gt"
        ]
      },
      {
        "required": [
          "$ge"
        ]
      },
      {
        "required": [
          "$lt"
        ]
      },
      {
        "required": [
          "$le"
        ]
      },
      {
        "required": [
          "$contains"
        ]
      },
      {
        "required": [
          "$starts-with"
        ]
      },
      {
        "required": [
          "$ends-with"
        ]
      },
      {
        "required": [
          "$regex"
        ]
      },
      {
        "required": [
          "$match"
        ]
```

```
        }
      ],
      "additionalProperties": false
    },
    "logicalExpression": {
      "type": "object",
      "properties": {
        "$and": {
          "type": "array",
          "minItems": 2,
          "items": {
            "$ref": "#/definitions/logicalExpression"
          }
        },
        "$match": {
          "type": "array",
          "minItems": 1,
          "items": {
            "$ref": "#/definitions/matchExpression"
          }
        },
        "$or": {
          "type": "array",
          "minItems": 2,
          "items": {
            "$ref": "#/definitions/logicalExpression"
          }
        },
        "$not": {
          "$ref": "#/definitions/logicalExpression"
        },
        "$eq": {
          "$ref": "#/definitions/equalityComparisonItems"
        },
        "$ne": {
          "$ref": "#/definitions/equalityComparisonItems"
        },
        "$gt": {
          "$ref": "#/definitions/orderedComparisonItems"
        },
        "$ge": {
          "$ref": "#/definitions/orderedComparisonItems"
        },
        "$lt": {
          "$ref": "#/definitions/orderedComparisonItems"
        },
        "$le": {
          "$ref": "#/definitions/orderedComparisonItems"
        },
        "$contains": {
          "$ref": "#/definitions/stringItems"
        },
```

```
      "$starts-with": {
        "$ref": "#/definitions/stringItems"
      },
      "$ends-with": {
        "$ref": "#/definitions/stringItems"
      },
      "$regex": {
        "$ref": "#/definitions/stringItems"
      },
      "$boolean": {
        "type": "boolean"
      },
      "$boolCast": {
        "$ref": "#/definitions/Value"
      }
    },
    "oneOf": [
      {
        "required": [
          "$and"
        ]
      },
      {
        "required": [
          "$or"
        ]
      },
      {
        "required": [
          "$not"
        ]
      },
      {
        "required": [
          "$eq"
        ]
      },
      {
        "required": [
          "$ne"
        ]
      },
      {
        "required": [
          "$gt"
        ]
      },
      {
        "required": [
          "$ge"
        ]
      },
```

```
        {
          "required": [
            "$lt"
          ]
        },
        {
          "required": [
            "$le"
          ]
        },
        {
          "required": [
            "$contains"
          ]
        },
        {
          "required": [
            "$starts-with"
          ]
        },
        {
          "required": [
            "$ends-with"
          ]
        },
        {
          "required": [
            "$regex"
          ]
        },
        {
          "required": [
            "$boolean"
          ]
        },
        {
          "required": [
            "$boolCast"
          ]
        },
        {
          "required": [
            "$match"
          ]
        }
      ],
      "additionalProperties": false
    }
  }
}
```

The provided JSON schema includes a *modelStringPattern* starting with a $, so that the syntax of model elements can be checked by the schema, e.g. $sm.idShort.

String, number and boolean are supported by JSON itself. Specific JSON Schema definitions are provided for datatypes Hex, DateTime and Time: *hexLiteralPattern*, *dateTimeLiteralPattern* and *timeLiteralPattern*. The DateTime and Time definitions use explicit regular expressions based on the AAS core meta verification functions MatchesXsDateTime and MatchesXsTime. These verification functions correspond to the XSD-based DateTime and Time data types defined by the IDTA 01001 Part 1 Metamodel [1], especially Primitive and Simple Data Types. For the generated AAS core meta verification definitions, see MatchesXsDateTime and MatchesXsTime. They are based on the XML Schema xs:dateTime and xs:time lexical representations defined in https://www.w3.org/TR/xmlschema-2/#dateTime and https://www.w3.org/TR/xmlschema-2/#time, and the ISO 8601 formats referenced by XML Schema in https://www.w3.org/TR/xmlschema-2/#isoformats. Accordingly, DateTime and Time literals include seconds, may include fractional seconds, and may include a timezone (Z or an offset from -14:00 to +14:00). The end-of-day notation 24:00:00 is allowed, with only zero fractional seconds. The day, month and year ranges in the BNF and JSON Schema are lexical constraints only. Special calendar cases, such as February 29 in non-leap years or the 31st day of a month with only 30 days, shall still be rejected.

If an incoming value is provided in an incomplete ISO-like form, it shall be completed before it is serialized as a DateTime or Time literal. If a Time value has no timezone information, it shall be interpreted as UTC and serialized with Z. For example, the Time value 12:00:00 shall be interpreted and serialized as 12:00:00Z.

Value is used for comparisons and *StringValue* is used for the specific string operations. For type safety and automatic code generation, each possible type is available as an own object:

- String values can be given by *$field*, $*strVal*, $*attribute* oder $*strCast*.
- Numerical values can be given by $*numVal* or $*numCast*.
- Hex values can be given by $*hexVal* or $*hexCast*.
- Bool values can be given by *$boolean* or $*boolCast*.
- DateTime values can be given by $*dateTimeVal* or $*dateTimeCast*.
- Time values can be given by $*timeVal* or $*timeCast*.

Casts convert values to the given type if possible. DateTime casts use string operands; Time casts use string or DateTime operands. Direct comparisons between two FieldIdentifier operands are not valid because the type of both fields would be implicit. To compare two fields, at least one operand shall use an explicit cast, e.g. compare *$field* to $*strCast*, $*numCast*, $*dateTimeCast*, or $*timeCast*.

Parts of DateTime values can be extracted as numbers:

- *$dayOfWeek* extracts the day of week as number, starting with 0 for Sunday.
- *$dayOfMonth*, *$month*, *$year* extract the related part as number.

Except for direct FieldIdentifier operands in typed comparisons, if a conversion or any other operation is invalid, the complete expression has to be treated as invalid. An error message shall be generated and the result of the complete expression becomes FALSE, so that the access is not allowed and/or the filter is empty.

To allow the use of different types in operations, arrays are used for parameters. *comparisonItems* is used for comparisons and *stringItems* is used for specific string operations.

logicalExpressions and *match* can be recursively nested.

Exchange of Access Rules

Business Partners may be interested to exchange Access Rules.

As the AAS is a central point for data access, there is the need to support fine-grained access control that supports multiple roles as well as separate access control policies for individual elements or submodels in the AAS. Access Control is based on Identity Management and can only be successfully implemented in a secure environment. For this

document, the focus lies on the supported access control model.

When AAS content is passed from one partner to another, this is typically related to a change in the access control domain of the partners involved (supplier, integrator, operator), i.e. the scope of the validity of access control policies.

In addition to the AAS content itself, also the defined access permissions have to be transferred between the partners due to the following two reasons:

1. Access permissions to information elements of an AAS must be established in each access control domain.
2. One partner must be able to pass a suggestion which access permissions should be established for the asset that is described in the AAS.

ABAC is a very flexible approach, that also encompasses role-based access as a role can be considered as one attribute in this context. Other attributes might be the time-of-day, the location of the asset, the originating address and others.

As explained as an example above, a robot manufacturer suggests that for the robot the following roles shall be defined: machine setter, operator and a maintenance role. Note that the roles have to be expressed by means of attributes of the AAS representing the robot. He also suggests permissions for these roles, e.g. an installer (integrator) does have write-access to the program of the robot, but an operator does not.

Providing such suggested access rules, makes the integration of the operator much easier. The operator can copy and paste the access rules into his AAS implementation and only needs to make adjustments.

The above example motivates that the access permission rules need to be passed from one access control domain to the other.

The enforcement of Access Rules is implementation specific.

This specification defines the text serialization of access rules and the JSON schema serialization of access rules. The support of both is optional and not mandatory.

The JSON schema serialization is recommended.

Such serialization provides the possibility to exchange access rules. An implementation may even store such serialized access rules in submodel elements BLOB or FILE, so that these can be accessed by the API and can be protected by access rules itself.

API Queries and Access Rules

Version 3.1 of the AAS API defines queries and uses the same grammar and same JSON schema to define queries. The JSON schema serialization is used in the related API operations.

In addition, using the same concepts will be needed for large amounts of AAS data. Querying such large amounts of data and protecting such data with Access Rules requires optimized access of data. Database queries may combine the filter expression provided by the client query with the FOMULAs of the Access Rules on the server.

Accountability for AAS Content and Digital Signatures

This document does not specify a native way to protect AAS elements with data-bound protection mechanisms such as digital signatures.

The AAS interface includes a basic way to protect the integrity and authenticity of information exchanged between the AAS user application and the AAS beyond the protection provided by the transport layer using the /$signed endpoint (IDTA 01002 [2]).

Apart from this accountability of information conveyed via AAS depends on the combination of several security measures:

- Access control ([access-rule-model:::access-rule-model]) to ensure that only authorized parties are allowed to provide and modify information or perform actions resulting in the change of information.
- Auditing (Auditable events, Timestamps, and Non-repudiation) to allow tracking of identities providing information into the AAS or performing actions resulting in the change of information.
- Data in transit protection (Information integrity and Session integrity) to ensure that integrity and authenticity of information is maintained during exchange of information between the AAS user application and the AAS. This applies to both, provisioning of information towards the AAS and requesting information from the AAS.

An AAS user application depending on a high assurance level regarding accountability and non-repudiation for individual elements of the AAS can digitally sign information stored as AAS payload, e.g., using a Property of type Blob.

Digital signatures provide a proven way to ensure the authenticity of digital information. The source of the data uses a (private) cryptographic key to create a digital signature on the digital information. The recipient uses a (public) cryptographic key to verify the signature. If the verification succeeds and the recipient knows the relation between the source and the public key, the recipient can conclude that the information has not been modified or tampered with since the creation of the signature. Digitally signing AAS payloads applies especially if information originating from a 3rd party (i.e., not the AAS Responsible) is relayed via AAS. Digital signing AAS payload information should also be considered if proof of authenticity is required outside the scope of the AAS context, e.g., when information is forwarded to a 3rd party which is not related to the AAS providing the information.

For example, it is possible to store a signed PDF or XML file as SubmodelElement value. The AAS Responsible shall define the syntax and semantic of such elements. It is also possible to store detached digital signatures of SubmodelElements in the value of a Property.

Secure Asset Binding

The association between AAS and asset depends on the correctness of the globalAssetId (or specificAssetIds) information stored in the AAS.

This association is unprotected and is subject to attack on the side of the AAS as well as on the side of the asset (e.g., tampered data carrier). IEEE 802.1AR [12] specifies Secure Device Identities designed to be used as interoperable secure device authentication credentials.

AAS could leverage these secure device identities to support the AAS user application to do secure device identification. This can be used by an AAS user application to verify whether the information gathered from the AAS applies to the asset (e.g., an automation device) in question. This could be achieved by either adding information on the device's IDevID certificate into the AAS information model, or by defining a Submodel for IEEE 802.1AR [12].

Summary and Outlook

This document specifies the Access Rule Model for the Asset Administration Shell APIs, including the APIs for repositories and registries.

The general concept of access tokens allows to combine the Access Rule Model with any available security infrastructure in companies, dataspaces or for legal requirements.

The grammar of the Access Rule Model allows to adopt the concepts to any further technology besides AAS HTTP API. For AAS HTTP API a JSON schema is already defined.

The grammar MAY also be used to create mappings to other access rule languages, e.g. XACML or ODRL.

This document is the base for the upcoming IEC 63278-3 “Security of the Asset Administration Shell”. IEC 63278-3 uses the explained concepts of access token together with the grammar for access rules.

Signing of Identifiables is defined in the REST API specification. This MAY be extended to SubmodelElements or other parts of Identifiables. Currently plain JWS (JSON Web Signature) is used for signing, which MAY be extended to additional formats e.g. JAdES (JSON Advanced Digital Signature). In addition AASX packages can be signed.

A next version of this document MAY also include an API to manage access rules. Since the grammar and the JSON schema are already used for the Query Language in AAS HTTP API 3.1, the needed elements for such additional API are already available.

Annex

Digital Product Passport

Introduction

To support the implementation of digital product passports (DPPs), 8 standards have been developed in CEN CENELEC:

- EN 18216:2026 – Digital product passport – Data exchange protocols
- EN 18219:2026 – Digital product passport – Unique identifiers
- EN 18220:2026 – Digital product passport – Data carriers
- EN 18221:2026 – Digital product passport – Data storage, archiving, and data persistence
- EN 18222:2026 – Digital Product Passport – Application Programming Interfaces (APIs) for the product passport lifecycle management and searchability
- EN 18223:2026 – Digital Product Passport – System interoperability
- EN 18239:2026 – Digital Product Passport – Access rights management, information system security, and business confidentiality
- EN 18246:2026 – Digital Product Passport – Data authentication, reliability and integrity

This annex specifies how the requirements as specified in

- EN 18239:2026 - Digital product passport - Access rights management, information system security and business confidentifality
- EN 18246:2026 – Digital Product Passport – Data authentication, reliability and integrity

are mapped or realized in the context of an Asset Administration Shell (AAS) system.

The two standards above are expected to be released in September 2026. Until then no final mapping and explanations can be given in public.

Assessment of IEC 62443 requirements for AAS structure

Table Table 12 ,Table 13, Table 14, Table 15, Table 16, Table 17 and Table 18 provide an assessment regarding the relevance of the foundation requirements defined by the IEC 62443 series towards the structure and interface of the AAS as defined in IEC 63278-1,-2,-5. Since AAS shares aspects of a system and a component both IEC 62443-3-3 and IEC 62443-4-2 have been considered.

According to the analysis not all foundational requirements given in IEC 62443 are applicable on the level of IEC 63278-1,-2,-5. “Y” means the requirement applies and is addressed within this document, “N” means the requirement does not apply. “P” means that the requirement is partially addressed by this document, details are either defined by other applicable standards or left to the implementation.

The full set of IEC 62443 requirements needs to be reconsidered for additional system components (e.g., AAS user application, infrastructure) and for subsequent realization steps (detailed design, implementation, system integration, operation).

Table 12. Assessment of IEC 62443 FR1 Authentication Control requirements for AAS

FR	Title	IEC 62443-3-3 SR	IEC 62443-4-2 CR	Applicable to IDTA 01001 , IDTA 01002 , IDTA 01003-a	IDTA 01004 clause or comment
1.1	Human user, process and device identification and authentication	¬	¬	Y	Identification and authentication of AAS user application users
1.2	Software process and device identification and authentication	¬	¬	Y	Identification and authentication of AAS user applications and AAS interface
1.3	Account management	¬	¬	Y	Account management
1.4	Identifier management	¬	¬	Y	Identifier management
1.5	Authenticator management	¬	¬	P	AAS authorization solely relies on short-term authentication tokens (Attributes). Measures for management and protection of access tokens are left to the implementation.
1.6	Wireless access management	¬	¬	N	IDTA 01001 [1], IDTA 01002 [2], IDTA 01003-a [3] do not make any assumption on network physical infrastructure. This is left to the implementation.
1.7	Strength of password-based authentication	¬	¬	N	AAS interface does not define password-based authentication.

FR	Title	IEC 62443-3-3 SR	IEC 62443-4-2 CR	Applicable to IDTA 01001 , IDTA 01002 , IDTA 01003-a	IDTA 01004 clause or comment
1.8	Public key infrastructure certificates	¬	¬	P	Public key infrastructure It is possible to use public key certificates to meet authentication requirements for AAS regarding, e.g., acquisition of an access tokens (Attributes), for TLS channel protection (Identification and authentication of AAS user application users, Identification and authentication of AAS user applications and AAS interface, Information integrity, Information confidentiality).
1.9	Strength of public key authentication	¬	¬	P	Use of cryptography
1.10	Authenticator feedback	¬	¬	Y	Authenticator feedback
1.11	Unsuccessful login attempts	¬	¬	N	AAS interface does not provide lock-out for users. This is left to the authentication service used to acquire the access token or the AAS user application.

FR	Title	IEC 62443-3-3 SR	IEC 62443-4-2 CR	Applicable to IDTA 01001 , IDTA 01002 , IDTA 01003-a	IDTA 01004 clause or comment
1.12	System use notification	¬	¬	N	This is left to the AAS user application.
1.13	Access via untrusted networks	¬	¬	N	IDTA 01001 [1], IDTA 01002 [2], IDTA 01003-a [3] do not make any assumption on network physical infrastructure. This is left to the implementation.
1.14	Strength of symmetric key authentication		¬	P	Use of cryptography

Table 13. Assessment of IEC 62443 FR2 Use Control requirements for AAS

FR	Title	IEC 62443-3-3 SR	IEC 62443-4-2 CR	Applicable to IDTA 01001 , IDTA 01002 , IDTA 01003-a	IDTA 01004 clause or comment
2.1	Authorization enforcement	¬	¬	Y	Authorization enforcement, Access Rule Model
2.2	Wireless use control	¬	¬	N	IDTA 01001 [1], IDTA 01002 [2], IDTA 01003-a [3] are hardware agnostic
2.3	Use control for portable and mobile devices	¬	¬	N	IDTA 01001 [1], IDTA 01002 [2], IDTA 01003-a [3] do not consider deployment aspects
2.4	Mobile code	¬	¬	N	IDTA 01001 [1], IDTA 01002 [2], IDTA 01003-a [3] do not consider deployment aspects

FR	Title	IEC 62443-3-3 SR	IEC 62443-4-2 CR	Applicable to IDTA 01001 , IDTA 01002 , IDTA 01003-a	IDTA 01004 clause or comment
2.5	Session lock	¬	¬	N	AAS does not define (direct) human access to the AAS interface. This is left to the AAS user application.
2.6	Remote session termination	¬	¬	Y	Limited life-time of access tokens. Attributes
2.7	Concurrent session control	¬	¬	N	AAS does not impose any limits on the amount and concurrency of requests from the AAS user application. This is left to the implementation.
2.8	Auditable events	¬	¬	Y	Auditable events
2.9	Audit storage capacity	¬	¬	N	AAS does not define audit storage.
2.10	Response to audit processing failures	¬	¬	N	AAS does not define audit storage.
2.11	Timestamps	¬	¬	Y	Timestamps
2.12	Non-repudiation	¬	¬	Y	Non-repudiation, Accountability for AAS Content and Digital Signatures
2.13	Use of physical diagnostic and test interfaces		¬	N	AAS is hardware agnostic.

Table 14. Assessment of IEC 62443 FR3 System Integrity requirements for AAS

FR	Title	IEC 62443-3-3 SR	IEC 62443-4-2 CR	Applicable to IDTA 01001 , IDTA 01002 , IDTA 01003-a	IDTA 01004 clause or comment
3.1	Communication integrity	¬	¬	Y	Information integrity
3.2	Malicious code protection	¬	¬	N	IDTA 01001 [1], IDTA 01002 [2], IDTA 01003-a [3] do not consider deployment and implementation specific security aspects.
3.3	Security functionality verification	¬	¬	N	IDTA 01001 [1], IDTA 01002 [2], IDTA 01003-a [3] do not consider deployment and implementation specific security aspects.
3.4	Software and information integrity	¬	¬	Y	Information integrity
3.5	Input validation	¬	¬	Y	Input validation
3.6	Deterministic output	¬	¬	N	AAS degradation behaviour is use case and implementation specific.
3.7	Error handling	¬	¬	Y	Error handling
3.8	Session integrity	¬	¬	Y	Session integrity
3.9	Protection of audit information	¬	¬	N	AAS does not define audit storage.
3.10	Support for updates		¬	N	IDTA 01001 [1], IDTA 01002 [2], IDTA 01003-a [3] do not consider operations aspects.

FR	Title	IEC 62443-3-3 SR	IEC 62443-4-2 CR	Applicable to IDTA 01001 , IDTA 01002 , IDTA 01003-a	IDTA 01004 clause or comment
3.11	Physical tamper resistance and detection		¬	P	Potential measures to protect integrity of data represented by the AAS include technologies for physical tamper resistance and detection. IDTA 01001 [1], IDTA 01002 [2], IDTA 01003-a [3] do not define hardware aspects including physical tamper resistance and detection. This is left to the implementation and operation based on the assessment of the associated risks. Information integrity
3.12	Provisioning product supplier roots of trust		¬	P	Entity (product) in question is the asset managed by the AAS. Asset roots of trust managment
3.13	Provisioning asset owner roots of trust		¬	Y	Entity (asset) in question is the AAS. AAS roots of trust management

FR	Title	IEC 62443-3-3 SR	IEC 62443-4-2 CR	Applicable to IDTA 01001 , IDTA 01002 , IDTA 01003-a	IDTA 01004 clause or comment
3.14	Integrity of the boot process		¬	N	IDTA 01001 [1], IDTA 01002 [2], IDTA 01003-a [3] do not define operation platform specific aspects including integrity of the boot process. This is left to the implementation and operation.

Table 15. Assessment of IEC 62443 FR4 Data confidentiality requirements for AAS

FR	Title	IEC 62443-3-3 SR	IEC 62443-4-2 CR	Applicable to IDTA 01001 , IDTA 01002 , IDTA 01003-a	IDTA 01004 clause or comment
4.1	Information confidentiality	¬	¬	Y	Information confidentiality
4.2	Information persistence	¬	¬	N	IDTA 01001 [1], IDTA 01002 [2], IDTA 01003-a [3] do not consider information persistence aspects. This is left to the operation and maintenance.
4.3	Use of cryptography	¬	¬	Y	Use of cryptography

Table 16. Assessment of IEC 62443 FR5 Restricted data flow requirements for AAS

FR	Title	IEC 62443-3-3 SR	IEC 62443-4-2 CR	Applicable to IDTA 01001 , IDTA 01002 , IDTA 01003-a	IDTA 01004 clause or comment
5.1	Network segmentation	¬	¬	N	IDTA 01001 [1], IDTA 01002 [2], IDTA 01003-a [3] do not consider deployment aspects.
5.2	Zone boundary protection	¬	¬	N	IDTA 01001 [1], IDTA 01002 [2], IDTA 01003-a [3] do not define network deployment and topology aspects.
5.3	General purpose person-to-person communication restrictions	¬	¬	N	IDTA 01001 [1], IDTA 01002 [2], IDTA 01003-a [3] do not consider person-to-person communication.
5.4	Application partitioning	¬	¬	Y	Application partitioning

Table 17. Assessment of IEC 62443 FR6 Timely response to events requirements for AAS

FR	Title	IEC 62443-3-3 SR	IEC 62443-4-2 CR	Applicable to IDTA 01001 , IDTA 01002 , IDTA 01003-a	IDTA 01004 clause or comment
6.1	Audit log accessibility	¬	¬	P	Audit log accessibility
6.2	Continuous monitoring	¬	¬	Y	Continuous monitoring

Table 18. Assessment of IEC 62443 FR7 Resource Availability requirements for AAS

FR	Title	IEC 62443-3-3 SR	IEC 62443-4-2 CR	Applicable to IDTA 01001 , IDTA 01002 , IDTA 01003-a	IDTA 01004 clause or comment
7.1	Denial of service protection	¬	¬	Y	Denial-of-service protection

FR	Title	IEC 62443-3-3 SR	IEC 62443-4-2 CR	Applicable to IDTA 01001 , IDTA 01002 , IDTA 01003-a	IDTA 01004 clause or comment
7.2	Resource management	¬	¬	N	IDTA 01001 [1], IDTA 01002 [2], IDTA 01003-a [3] do not define resource deployment aspects.
7.3	Control system backup	¬	¬	N	IDTA 01001 [1], IDTA 01002 [2], IDTA 01003-a [3] do not define resource deployment and operations aspects.
7.4	Control system recovery and reconstitution	¬	¬	Y	AAS recovery and reconstitution
7.5	Emergency power	¬	¬	N	IDTA 01001 [1], IDTA 01002 [2], IDTA 01003-a [3] do not define network deployment aspects.
7.6	Network and security configuration settings	¬	¬	N	IDTA 01001 [1], IDTA 01002 [2], IDTA 01003-a [3] do not define network deployment aspects.
7.7	Least functionality	¬	¬	Y	Least functionality
7.8	Control system component inventory	¬	¬	N	IDTA 01001 [1], IDTA 01002 [2], IDTA 01003-a [3] do not define deployment aspects.

Backus-Naur-Form

The Backus-Naur form (BNF) – a meta-syntax notation for context-free grammars – is used to define grammars. For more information see Wikipedia.

A BNF specification is a set of derivation rules, written as

```
<symbol> ::= __expression__
```

where:

- <symbol> is a nonterminal (variable) and the *expression* consists of one or more sequences of either terminal or nonterminal symbols,
- ::= means that the symbol on the left MUST be replaced with the expression on the right,
- more sequences of symbols are separated by the vertical bar "|", indicating a choice, the whole being a possible substitution for the symbol on the left,
- symbols that never appear on a left side are terminals, while symbols that appear on a left side are non-terminals and are always enclosed between the pair of angle brackets <>,
- terminals are enclosed with quotation marks: "text". "" is an empty string,
- OPTIONAL items are enclosed in square brackets: [<item-x>],
- items existing 0 or more times are enclosed in round brackets and suffixed with an asterisk (*) such as <word> ::= <letter> (<letter>)*,
- Items existing 1 or more times are suffixed with an addition (plus) symbol, +, such as <word> ::= (<letter>)+,
- round brackets are also used to explicitly define the order of expansion to indicate precedence, example: (<symbol1> | <symbol2>) <symbol3>,
- text without quotation marks is an informal explanation of what is expected; this text is cursive if grammar is non-recursive and vice versa.

Example:

```
<contact-address> ::= <name> "e-mail addresses:" <e-mail-Addresses>

<e-mail-Addresses> ::= (<e-mail-Address>)*

<e-mail-Address> ::= <local-part> "@" <domain>

<name> ::= characters

<local-part> ::= characters conformant to local-part in RFC 5322

<domain> ::= characters conformant to domain in RFC 5322
```

Valid contact addresses:

```
Hugo Me e-mail addresses: Hugo@example.com

Hugo e-mail addresses: Hugo.Me@text.de
```

Invalid contact addresses:

```
Hugo
```

```
Hugo Hugo@ example.com

Hugo@example.com
```

Examples of Access Rules in text serialization

Allow READ access for Anonymous to complete API

```
ACCESSRULE:
  ATTRIBUTES:
    GLOBAL(ANONYMOUS)
  RIGHTS: READ
  ACCESS: ALLOW
  OBJECTS:
    ROUTE "*"
  FORMULA:
    true
```

Allow READ access for Anonymous to list of semanticIDs for submodels

```
ACCESSRULE:
  ATTRIBUTES:
    GLOBAL(ANONYMOUS)
  RIGHTS: READ
  ACCESS: ALLOW
  OBJECTS:
    ROUTE  "*"
  FORMULA:
    $or(
      $sm#semanticId $eq "SemanticID-Nameplate",
      $sm#semanticId $eq "SemanticID-TechnicalData"
    )
```

Allow EXECUTE of API operations only if machine not-running

```
ACCESSRULE:
  ATTRIBUTES:
    GLOBAL(ANONYMOUS)
  RIGHTS: EXECUTE
  ACCESS: ALLOW
  OBJECTS:
    ROUTE  "*"
  FORMULA:
    REFERENCE($sme("SubmodelID-OperationalData").machineState#value) $eq "not-running"
```

Allow READ and UPDATE for specific authenticated users

```
ACCESSRULE:
  ATTRIBUTES:
    CLAIM("email")
  RIGHTS: READ UPDATE
  ACCESS: ALLOW
  OBJECTS:
    IDENTIFIABLE $sm("*")
  FORMULA:
    $and(
      $or(
        $sm#semanticId $eq "SemanticID-Nameplate",
        $sm#semanticId $eq "SemanticID-TechnicalData"
      ),
      $or(
        CLAIM("email") $eq "user1@company1.com",
        CLAIM("email") $eq "user2@company2.com"
      )
    )
```

Allow READ and UPDATE for specific submodel "submodel1"

```
ACCESSRULE:
  ATTRIBUTES:
    CLAIM("email")
  RIGHTS: READ UPDATE
  ACCESS: ALLOW
  OBJECTS:
    IDENTIFIABLE $sm("https://submodel1.company1.com")
  FORMULA:
    CLAIM("email") $eq "user1@company1.com"
```

Reuse of ACL, OBJECT and FORMULA

```
DEFACLS "acl1"
  ATTRIBUTES:
    CLAIM("email")
  RIGHTS: READ UPDATE
  ACCESS: ALLOW

DEFOBJECTS "Properties"
  REFERABLE $sme("https://s1.com").p1
  REFERABLE $sme("https://s1.com").p2

DEFFORMULAS "allowSubjectGroup1"
  $and(
    time(GLOBAL(UTCNOW)) $eq 15:00:00Z,
```

```
    $or(
      CLAIM("email") $eq "user1@company1.com",
      CLAIM("email") $eq "user2@company2.com"
    )
  )

ACCESSRULE:
  USEACL "acl1"
  OBJECTS:
    USEOBJECTS "Properties"
  USEFORMULA "allowSubjectGroup1"
```

Example with BusinessPartnerNumber

```
ACCESSRULE:
  ATTRIBUTES:
   CLAIM("BusinessPartnerNumber")
  RIGHTS: READ
  ACCESS: ALLOW
  OBJECTS:
    ROUTE "*"
  FORMULA:
    CLAIM("BusinessPartnerNumber") $eq "BPN1234"
```

Allow READ for all authenticated users of a company for submodels Nameplate and TechnicalData

```
ACCESSRULE:
  ATTRIBUTES:
  CLAIM("email")
  RIGHTS: READ
  ACCESS: ALLOW
  OBJECTS:
    IDENTIFIABLE $sm("*")
  FORMULA:
    $and(
      $or(
        $sm#semanticId $eq "SemanticID-Nameplate",
        $sm#semanticId $eq "SemanticID-TechnicalData"
      ),
      $regex(CLAIM("email"), "[\w\.]+@company\.com")
    )
```

Allow READ to all Submodels with ID pattern for all authenticated users of a company for submodels with Nameplate and TechnicalData on weekdays from 09:00:00Z-17:00:00Z

```
ACCESSRULE:
  ATTRIBUTES:
  CLAIM("companyName")
  RIGHTS: READ
  ACCESS: ALLOW
  OBJECTS:
    IDENTIFIABLE $sm("*")
  FORMULA:
    $and(
      $or(
        $sm#semanticId $eq "SemanticID-Nameplate",
        $sm#semanticId $eq "SemanticID-TechnicalData"
      ),
      CLAIM("companyName") $eq "company1-name",
      $regex($sm#id, "^https://company1.com/.*$"),
      $dayOfWeek(GLOBAL(UTCNOW)) $ge 1,
      $dayOfWeek(GLOBAL(UTCNOW)) $le 5,
      time(GLOBAL(UTCNOW)) $ge 09:00:00Z,
      time(GLOBAL(UTCNOW)) $le 17:00:00Z
    )
```

Allow only to add elements to the CertificateSet in any Submodel

```
ACCESSRULE:
  ATTRIBUTES:
    CLAIM("Role")
  RIGHTS: CREATE
  ACCESS: ALLOW
  OBJECTS:
    IDENTIFIABLE $sm("*")
  FORMULA:
    $and(
      CLAIM("Role") $eq "person with legitimate interest",
      $sme#semanticId $eq "CertificateSet"
    )
```

Example with FILTER statement

```
ACCESSRULE:
  ATTRIBUTES:
    CLAIM("BusinessPartnerNumber")
  RIGHTS: READ
  ACCESS: ALLOW
  OBJECTS:
```

```
    DESCRIPTOR $aasdesc("*")
  FORMULA:
    $and(
      CLAIM("BusinessPartnerNumber") $eq "BPNL00000000000A",
      $match(
        $aasdesc#specificAssetIds[].name  $eq "manufacturerPartId",
        $aasdesc#specificAssetIds[].value $eq "99991",
        $aasdesc#specificAssetIds[].externalSubjectId $eq "PUBLIC_READABLE"
      ),
      $match(
        $aasdesc#specificAssetIds[].name $eq "customerPartId",
        $aasdesc#specificAssetIds[].value $eq "ACME001"
      )
    )
  FILTER:
    FRAGMENT: $aasdesc#specificAssetIds[]
    CONDITION:
    $or(
      $match(
        $aasdesc#specificAssetIds[].name  $eq "manufacturerPartId",
        $aasdesc#specificAssetIds[].value $eq "99991"
      ),
      $match(
        $aasdesc#specificAssetIds[].name $eq "customerPartId",
        $aasdesc#specificAssetIds[].value $eq "ACME001"
      ),
      $aasdesc#specificAssetIds[].name $eq "partInstanceId",
      $aasdesc#specificAssetIds[].externalSubjectId $eq CLAIM("BusinessPartnerNumber"),
      $aasdesc#specificAssetIds[].externalSubjectId $eq "PUBLIC_READABLE"
    )
```

Example with Reference Attribute and state-dependent filtering

```
DEFACLS "maintenanceRead"
  ATTRIBUTES:
    CLAIM("role")
  RIGHTS: READ
  ACCESS: ALLOW

DEFOBJECTS "MaintenanceDocuments"
  REFERABLE $sme("SubmodelID-Maintenance").maintenanceDocuments

DEFFORMULAS "maintenanceWhenRunning"
  $and(
    CLAIM("role") $eq "maintenance",
    REFERENCE($sme("SubmodelID-OperationalData").machineState#value) $eq "running"
  )

DEFFORMULAS "maintenanceWhenNotRunning"
  $and(
```

```
    CLAIM("role") $eq "maintenance",
    REFERENCE($sme("SubmodelID-OperationalData").machineState#value) $eq "not-running"
  )

DEFFORMULAS "matchingMaintenanceDocuments"
  $sme.maintenanceDocuments[].requiredMachineState#value $eq REFERENCE($sme("SubmodelID-
OperationalData").machineState#value)

ACCESSRULE:
  USEACL "maintenanceRead"
  OBJECTS:
    USEOBJECTS "MaintenanceDocuments"
  USEFORMULA "maintenanceWhenRunning"
  FILTER:
    FRAGMENT: $sme.maintenanceDocuments[]
    USEFORMULA "matchingMaintenanceDocuments"

ACCESSRULE:
  USEACL "maintenanceRead"
  OBJECTS:
    USEOBJECTS "MaintenanceDocuments"
  USEFORMULA "maintenanceWhenNotRunning"
```

Examples of Access Rules in JSON serialization

Allow READ access for Anonymous to complete API

```
{
  "AllAccessPermissionRules": {
    "rules": [
      {
        "ACL": {
          "ATTRIBUTES": [
            {
              "GLOBAL": "ANONYMOUS"
            }
          ],
          "RIGHTS": [
            "READ"
          ],
          "ACCESS": "ALLOW"
        },
        "OBJECTS": [
          {
            "ROUTE": "*"
          }
        ],
        "FORMULA": {
          "$boolean": true
        }
      }
```

```
    ]
  }
}
```

Allow READ access for Anonymous to list of semanticIDs for submodels

```
{
  "AllAccessPermissionRules": {
    "rules": [
      {
        "ACL": {
          "ATTRIBUTES": [
            {
              "GLOBAL": "ANONYMOUS"
            }
          ],
          "RIGHTS": [
            "READ"
          ],
          "ACCESS": "ALLOW"
        },
        "OBJECTS": [
          {
            "ROUTE": "*"
          }
        ],
        "FORMULA": {
          "$or": [
            {
              "$eq": [
                {
                  "$field": "$sm#semanticId"
                },
                {
                  "$strVal": "SemanticID-Nameplate"
                }
              ]
            },
            {
              "$eq": [
                {
                  "$field": "$sm#semanticId"
                },
                {
                  "$strVal": "SemanticID-TechnicalData"
                }
              ]
            }
          ]
        }
```

```
      }
    ]
  }
}
```

Allow EXECUTE of API operations only if machine not-running

```
{
  "AllAccessPermissionRules": {
    "rules": [
      {
        "ACL": {
          "ATTRIBUTES": [
            {
              "GLOBAL": "ANONYMOUS"
            }
          ],
          "RIGHTS": [
            "EXECUTE"
          ],
          "ACCESS": "ALLOW"
        },
        "OBJECTS": [
          {
            "ROUTE": "*"
          }
        ],
        "FORMULA": {
          "$eq": [
            {
              "$attribute": {
                "REFERENCE": "$sme(\"SubmodelID-OperationalData\").machineState#value"
              }
            },
            {
              "$strVal": "not-running"
            }
          ]
        }
      }
    ]
  }
}
```

Allow READ and UPDATE for specific authenticated users

```
{
  "AllAccessPermissionRules": {
    "rules": [
```

```
{
  "ACL": {
    "ATTRIBUTES": [
      {
        "CLAIM": "email"
      }
    ],
    "RIGHTS": [
      "READ",
      "UPDATE"
    ],
    "ACCESS": "ALLOW"
  },
  "OBJECTS": [
    {
      "IDENTIFIABLE": "$sm(\"*\")"
    }
  ],
  "FORMULA": {
    "$and": [
      {
        "$or": [
          {
            "$eq": [
              {
                "$field": "$sm#semanticId"
              },
              {
                "$strVal": "SemanticID-Nameplate"
              }
            ]
          },
          {
            "$eq": [
              {
                "$field": "$sm#semanticId"
              },
              {
                "$strVal": "SemanticID-TechnicalData"
              }
            ]
          }
        ]
      },
      {
        "$or": [
          {
            "$eq": [
              {
                "$attribute": {
                  "CLAIM": "email"
                }
```

```
                    },
                    {
                      "$strVal": "user1@company1.com"
                    }
                  ]
                },
                {
                  "$eq": [
                    {
                      "$attribute": {
                        "CLAIM": "email"
                      }
                    },
                    {
                      "$strVal": "user2@company2.com"
                    }
                  ]
                }
              ]
            }
          ]
        }
      }
    ]
  }
}
```

Allow READ and UPDATE for specific submodel "submodel1"

```
{
  "AllAccessPermissionRules": {
    "rules": [
      {
        "ACL": {
          "ATTRIBUTES": [
            {
              "CLAIM": "email"
            }
          ],
          "RIGHTS": [
            "READ",
            "UPDATE"
          ],
          "ACCESS": "ALLOW"
        },
        "OBJECTS": [
          {
            "IDENTIFIABLE": "$sm(\"https://submodel1.company1.com\")"
          }
        ],
```

```
        "FORMULA": {
          "$eq": [
            {
              "$attribute": {
                "CLAIM": "email"
              }
            },
            {
              "$strVal": "user1@company1.com"
            }
          ]
        }
      }
    ]
  }
}
```

Reuse of ACL, OBJECT and FORMULA

```
{
  "AllAccessPermissionRules": {
    "DEFACLS": [
      {
        "name": "acl1",
        "acl": {
          "ATTRIBUTES": [
            {
              "CLAIM": "email"
            }
          ],
          "RIGHTS": [
            "READ",
            "UPDATE"
          ],
          "ACCESS": "ALLOW"
        }
      }
    ],
    "DEFOBJECTS": [
      {
        "name": "Properties",
        "objects": [
          {
            "REFERABLE": "$sme(\"https://s1.com\").p1"
          },
          {
            "REFERABLE": "$sme(\"https://s1.com\").p2"
          }
        ]
      }
```

```
  ],
  "DEFFORMULAS": [
    {
      "name": "allowSubjectGroup1",
      "formula": {
        "$and": [
          {
            "$eq": [
              {
                "$timeCast": {
                  "$attribute": {
                    "GLOBAL": "UTCNOW"
                  }
                }
              },
              {
                "$timeVal": "15:00:00Z"
              }
            ]
          },
          {
            "$or": [
              {
                "$eq": [
                  {
                    "$attribute": {
                      "CLAIM": "email"
                    }
                  },
                  {
                    "$strVal": "user1@company1.com"
                  }
                ]
              },
              {
                "$eq": [
                  {
                    "$attribute": {
                      "CLAIM": "email"
                    }
                  },
                  {
                    "$strVal": "user2@company2.com"
                  }
                ]
              }
            ]
          }
        ]
      }
    }
  ],
```

```
    "rules": [
      {
        "USEACL": "acl1",
        "USEOBJECTS": [ "Properties" ],
        "USEFORMULA": "allowSubjectGroup1"
      }
    ]
  }
}
```

Example with BusinessPartnerNumber

```
{
  "AllAccessPermissionRules": {
    "rules": [
      {
        "ACL": {
          "ATTRIBUTES": [
            {
              "CLAIM": "BusinessPartnerNumber"
            }
          ],
          "RIGHTS": [
            "READ"
          ],
          "ACCESS": "ALLOW"
        },
        "OBJECTS": [
          {
            "ROUTE": "*"
          }
        ],
        "FORMULA": {
          "$eq": [
            {
              "$attribute": {
                "CLAIM": "BusinessPartnerNumber"
              }
            },
            {
              "$strVal": "BPN1234"
            }
          ]
        }
      }
    ]
  }
}
```

Allow READ for all authenticated users of a company for submodels Nameplate and TechnicalData

```
{
  "AllAccessPermissionRules": {
    "rules": [
      {
        "ACL": {
          "ATTRIBUTES": [
            {
              "CLAIM": "email"
            }
          ],
          "RIGHTS": [
            "READ"
          ],
          "ACCESS": "ALLOW"
        },
        "OBJECTS": [
          {
            "IDENTIFIABLE": "$sm(\"*\")"
          }
        ],
        "FORMULA": {
          "$and": [
            {
              "$or": [
                {
                  "$eq": [
                    {
                      "$field": "$sm#semanticId"
                    },
                    {
                      "$strVal": "SemanticID-Nameplate"
                    }
                  ]
                },
                {
                  "$eq": [
                    {
                      "$field": "$sm#semanticId"
                    },
                    {
                      "$strVal": "SemanticID-TechnicalData"
                    }
                  ]
                }
              ]
            },
            {
              "$regex": [
```

```
                {
                  "$attribute": {
                    "CLAIM": "email"
                  }
                },
                {
                  "$strVal": "[\\w\\.]+@company\\.com"
                }
              ]
            }
          ]
        }
      }
    ]
  }
}
```

Allow READ to all Submodels with ID pattern for all authenticated users of a company for submodels with Nameplate and TechnicalData on weekdays from 09:00:00Z-17:00:00Z

```
{
  "AllAccessPermissionRules": {
    "rules": [
      {
        "ACL": {
          "ATTRIBUTES": [
            {
              "CLAIM": "companyName"
            }
          ],
          "RIGHTS": [
            "READ"
          ],
          "ACCESS": "ALLOW"
        },
        "OBJECTS": [
          {
            "IDENTIFIABLE": "$sm(\"*\")"
          }
        ],
        "FORMULA": {
          "$and": [
            {
              "$or": [
                {
                  "$eq": [
                    {
                      "$field": "$sm#semanticId"
                    },
```

```
            {
              "$strVal": "SemanticID-Nameplate"
            }
          ]
        },
        {
          "$eq": [
            {
              "$field": "$sm#semanticId"
            },
            {
              "$strVal": "SemanticID-TechnicalData"
            }
          ]
        }
      ]
    },
    {
      "$eq": [
        {
          "$attribute": {
            "CLAIM": "companyName"
          }
        },
        {
          "$strVal": "company1-name"
        }
      ]
    },
    {
      "$regex": [
        {
          "$field": "$sm#id"
        },
        {
          "$strVal": "^https://company1.com/.*$"
        }
      ]
    },
    {
      "$ge": [
        {
          "$dayOfWeek": {
            "$attribute": {
              "GLOBAL": "UTCNOW"
            }
          }
        },
        {
          "$numVal": 1
        }
      ]
```

```
            },
            {
              "$le": [
                {
                  "$dayOfWeek": {
                    "$attribute": {
                      "GLOBAL": "UTCNOW"
                    }
                  }
                },
                {
                  "$numVal": 5
                }
              ]
            },
            {
              "$ge": [
                {
                  "$timeCast": {
                    "$attribute": {
                      "GLOBAL": "UTCNOW"
                    }
                  }
                },
                {
                  "$timeVal": "09:00:00Z"
                }
              ]
            },
            {
              "$le": [
                {
                  "$timeCast": {
                    "$attribute": {
                      "GLOBAL": "UTCNOW"
                    }
                  }
                },
                {
                  "$timeVal": "17:00:00Z"
                }
              ]
            }
          ]
        }
      }
    ]
  }
}
```

Example with FILTER statement

```
{
  "AllAccessPermissionRules": {
    "rules": [
      {
        "ACL": {
          "ATTRIBUTES": [
            {
              "CLAIM": "BusinessPartnerNumber"
            }
          ],
          "RIGHTS": [
            "READ"
          ],
          "ACCESS": "ALLOW"
        },
        "OBJECTS": [
          {
            "DESCRIPTOR": "$aasdesc(\"*\")"
          }
        ],
        "FORMULA": {
          "$and": [
            {
              "$eq": [
                {
                  "$attribute": {
                    "CLAIM": "BusinessPartnerNumber"
                  }
                },
                {
                  "$strVal": "BPNL00000000000A"
                }
              ]
            },
            {
              "$match": [
                {
                  "$eq": [
                    {
                      "$field": "$aasdesc#specificAssetIds[].name"
                    },
                    {
                      "$strVal": "manufacturerPartId"
                    }
                  ]
                },
                {
                  "$eq": [
                    {
                      "$field": "$aasdesc#specificAssetIds[].value"
                    },
```

```
              {
                "$strVal": "99991"
              }
            ]
          },
          {
            "$eq": [
              {
                "$field": "$aasdesc#specificAssetIds[].externalSubjectId"
              },
              {
                "$strVal": "PUBLIC_READABLE"
              }
            ]
          }
        ]
      },
      {
        "$match": [
          {
            "$eq": [
              {
                "$field": "$aasdesc#specificAssetIds[].name"
              },
              {
                "$strVal": "customerPartId"
              }
            ]
          },
          {
            "$eq": [
              {
                "$field": "$aasdesc#specificAssetIds[].value"
              },
              {
                "$strVal": "ACME001"
              }
            ]
          }
        ]
      }
    ]
  },
  "FILTER": {
    "FRAGMENT": "$aasdesc#specificAssetIds[]",
    "CONDITION": {
      "$or": [
        {
          "$match": [
            {
              "$eq": [
                {
```

```
                "$field": "$aasdesc#specificAssetIds[].name"
              },
              {
                "$strVal": "manufacturerPartId"
              }
            ]
          },
          {
            "$eq": [
              {
                "$field": "$aasdesc#specificAssetIds[].value"
              },
              {
                "$strVal": "99991"
              }
            ]
          }
        ]
      },
      {
        "$match": [
          {
            "$eq": [
              {
                "$field": "$aasdesc#specificAssetIds[].name"
              },
              {
                "$strVal": "customerPartId"
              }
            ]
          },
          {
            "$eq": [
              {
                "$field": "$aasdesc#specificAssetIds[].value"
              },
              {
                "$strVal": "ACME001"
              }
            ]
          }
        ]
      },
      {
        "$eq": [
          {
            "$field": "$aasdesc#specificAssetIds[].name"
          },
          {
            "$strVal": "partInstanceId"
          }
        ]
```

```
            },
            {
              "$eq": [
                {
                  "$field": "$aasdesc#specificAssetIds[].externalSubjectId"
                },
                {
                  "$attribute": {
                    "CLAIM": "BusinessPartnerNumber"
                  }
                }
              ]
            },
            {
              "$eq": [
                {
                  "$field": "$aasdesc#specificAssetIds[].externalSubjectId"
                },
                {
                  "$strVal": "PUBLIC_READABLE"
                }
              ]
            }
          ]
        }
      }
    ]
  }
}
```

Example with Reference Attribute and state-dependent filtering

```
{
  "AllAccessPermissionRules": {
    "DEFACLS": [
      {
        "name": "maintenanceRead",
        "acl": {
          "ATTRIBUTES": [
            {
              "CLAIM": "role"
            }
          ],
          "RIGHTS": [
            "READ"
          ],
          "ACCESS": "ALLOW"
        }
      }
```

```
  ],
  "DEFOBJECTS": [
    {
      "name": "MaintenanceDocuments",
      "objects": [
        {
          "REFERABLE": "$sme(\"SubmodelID-Maintenance\").maintenanceDocuments"
        }
      ]
    }
  ],
  "DEFFORMULAS": [
    {
      "name": "maintenanceWhenRunning",
      "formula": {
        "$and": [
          {
            "$eq": [
              {
                "$attribute": {
                  "CLAIM": "role"
                }
              },
              {
                "$strVal": "maintenance"
              }
            ]
          },
          {
            "$eq": [
              {
                "$attribute": {
                  "REFERENCE": "$sme(\"SubmodelID-OperationalData\").machineState#value"
                }
              },
              {
                "$strVal": "running"
              }
            ]
          }
        ]
      }
    },
    {
      "name": "maintenanceWhenNotRunning",
      "formula": {
        "$and": [
          {
            "$eq": [
              {
                "$attribute": {
                  "CLAIM": "role"
```

```
              }
            },
            {
              "$strVal": "maintenance"
            }
          ]
        },
        {
          "$eq": [
            {
              "$attribute": {
                "REFERENCE": "$sme(\"SubmodelID-OperationalData\").machineState#value"
              }
            },
            {
              "$strVal": "not-running"
            }
          ]
        }
      ]
    }
  },
  {
    "name": "matchingMaintenanceDocuments",
    "formula": {
      "$eq": [
        {
          "$field": "$sme.maintenanceDocuments[].requiredMachineState#value"
        },
        {
          "$attribute": {
            "REFERENCE": "$sme(\"SubmodelID-OperationalData\").machineState#value"
          }
        }
      ]
    }
  }
],
"rules": [
  {
    "USEACL": "maintenanceRead",
    "USEOBJECTS": [
      "MaintenanceDocuments"
    ],
    "USEFORMULA": "maintenanceWhenRunning",
    "FILTER": {
      "FRAGMENT": "$sme.maintenanceDocuments[]",
      "USEFORMULA": "matchingMaintenanceDocuments"
    }
  },
  {
    "USEACL": "maintenanceRead",
```

```
        "USEOBJECTS": [
          "MaintenanceDocuments"
        ],
        "USEFORMULA": "maintenanceWhenNotRunning"
      }
    ]
  }
}
```

Allow only to add elements to the CertificateSet in any Submodel

```
{
  "AllAccessPermissionRules": {
    "rules": [
      {
        "ACL": {
          "ATTRIBUTES": [
            {
              "CLAIM": "Role"
            }
          ],
          "RIGHTS": [
            "CREATE"
          ],
          "ACCESS": "ALLOW"
        },
        "OBJECTS": [
          {
            "IDENTIFIABLE": "$sm(\"*\")"
          }
        ],
        "FORMULA": {
          "$and": [
            {
              "$eq": [
                {
                  "$attribute": {
                    "CLAIM": "Role"
                  }
                },
                {
                  "$strVal": "person with legitimate interest"
                }
              ]
            },
            {
              "$eq": [
                {
                  "$field": "$sme#semanticId"
                },
```

```
                {
                  "$strVal": "CertificateSet"
                }
              ]
            }
          ]
        }
      }
    ]
  }
}
```

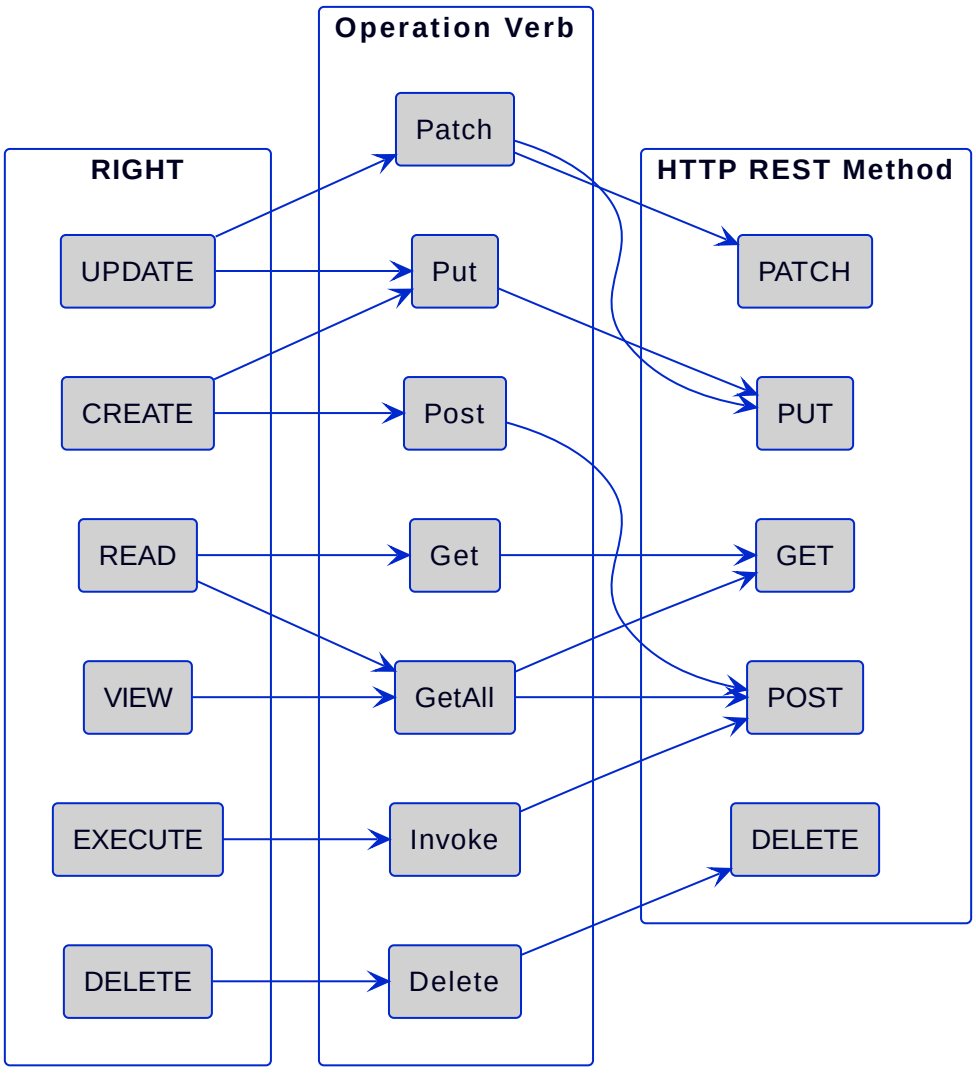
Examples of ROUTEs and HTTP requests

This annex is informative. It gives concrete examples for RIGHTS and ROUTE objects using AAS Part 2 HTTP/REST path templates.

Unless stated otherwise, the examples use the server base URL https://admin-shell.io/api/v3/ as published in the AAS Part 2 OpenAPI descriptions. Only the URL path is relevant for ROUTE matching.

Rights, operation verbs and HTTP methods

The following diagram illustrates one possible relation between abstract rights, operation verbs and HTTP methods. It is an example to support the interpretation of the normative mapping in Table 7.



Example requests by right

The following requests use real path templates from the published AAS Part 2 OpenAPI descriptions. They are representative examples for access control decisions on ROUTE objects. For client-addressable routes with create-or-replace or upsert semantics, the REQUIRED right depends on whether the addressed resource already exists. In the examples below, PUT /api/v3/submodels/{id} and POST /api/v3/lookup/shells/{someencodedid} therefore appear under both CREATE and UPDATE with different preconditions.

CREATE

Here are endpoints that can need CREATE rights to create new resources or entries in the repository or registry. PUT endpoints with create-or-replace semantics can require UPDATE instead of CREATE rights if the addressed resource already exists.

Example 1: create a new AAS in the repository by POST.

```
POST /api/v3/shells
```

Example 2: create a new submodel at a route by PUT if the addressed submodel does not yet exist.

```
PUT /api/v3/submodels/aHR0cHM6Ly9leGFtcGxlLmNvbS9zbS9uYW1lcGxhdGU
```

Example 3: create a new submodel element in a collection by POST.

```
POST /api/v3/submodels/aHR0cHM6Ly9leGFtcGxlLmNvbS9zbS9uYW1lcGxhdGU/submodel-elements
```

Example 4: create a lookup entry by POST if the addressed entry does not yet exist.

```
POST /api/v3/lookup/shells/someencodedid
```

READ

Here are endpoints that can need READ rights to access existing resources or entries in the repository or registry. VIEW right can be used as an alternative for GetAll endpoints if only existence information, references, identifiers, or descriptor entries SHALL be accessible.

Example 1: read all AAS objects.

```
GET /api/v3/shells?limit=50
```

Example 2: read one submodel.

```
GET /api/v3/submodels/aHR0cHM6Ly9leGFtcGxlLmNvbS9zbS9uYW1lcGxhdGU
```

Example 3: read one submodel element by path.

```
GET /api/v3/submodels/aHR0cHM6Ly9leGFtcGxlLmNvbS9zbS9uYW1lcGxhdGU/submodel-
```

```
elements/Identification/ManufacturerName
```

Example 4: read all shell descriptors from the registry.

```
GET /api/v3/shell-descriptors
```

Example 5: resolve AAS IDs through discovery.

This is an example of a read-like search operation implemented as POST.

```
POST /api/v3/lookup/shellsByAssetLink
```

UPDATE

Here are endpoints that can need UPDATE rights to modify existing resources or entries in the repository or registry. For client-addressable routes with create-or-replace semantics, CREATE rights can be sufficient if the addressed resource does not yet exist.

Example 1: replace an existing AAS resource by PUT.

```
PUT /api/v3/shells/aHR0cHM6Ly9leGFtcGxlLmNvbS9hYXMvMTIzNA
```

Example 2: replace an existing submodel at a client-addressable route by PUT if the addressed submodel already exists.

```
PUT /api/v3/submodels/aHR0cHM6Ly9leGFtcGxlLmNvbS9zbS9uYW1lcGxhdGU
```

Example 3: partially update a submodel by PATCH.

```
PATCH /api/v3/submodels/aHR0cHM6Ly9leGFtcGxlLmNvbS9zbS9uYW1lcGxhdGU
```

Example 4: update the value-only representation of a submodel element.

```
PATCH /api/v3/submodels/aHR0cHM6Ly9leGFtcGxlLmNvbS9zbS9uYW1lcGxhdGU/submodel-
elements/Identification/ManufacturerName/$value
```

Example 5: update an AAS descriptor in the registry.

```
PUT /api/v3/shell-descriptors/aHR0cHM6Ly9leGFtcGxlLmNvbS9hYXMvMTIzNA
```

Example 6: update a lookup entry by POST if the addressed entry already exists.

```
POST /api/v3/lookup/shells/someencodedid
```

DELETE

Here are endpoints that can need DELETE rights to delete existing resources or entries in the repository or registry.

Example 1: delete a submodel from the repository.

```
DELETE /api/v3/submodels/aHR0cHM6Ly9leGFtcGxlLmNvbS9zbS9uYW1lcGxhdGU
```

Example 2: delete a submodel element by path.

```
DELETE /api/v3/submodels/aHR0cHM6Ly9leGFtcGxlLmNvbS9zbS9uYW1lcGxhdGU/submodel-
elements/Identification/ManufacturerName
```

Example 3: delete a shell descriptor from the registry.

```
DELETE /api/v3/shell-descriptors/aHR0cHM6Ly9leGFtcGxlLmNvbS9hYXMvMTIzNA
```

EXECUTE

Here are endpoints that can need EXECUTE rights to invoke operations on the repository or registry.

Example 1: synchronously invoke an Operation.

```
POST /api/v3/submodels/aHR0cHM6Ly9leGFtcGxlLmNvbS9zbS9tYWludGVuYW5jZQ/submodel-
elements/Reset/invoke
```

Example 2: asynchronously invoke an Operation.

```
POST /api/v3/submodels/aHR0cHM6Ly9leGFtcGxlLmNvbS9zbS9tYWludGVuYW5jZQ/submodel-
elements/Reset/invoke-async
```

Example 3: retrieve the status of an asynchronous invocation.

```
GET /api/v3/submodels/aHR0cHM6Ly9leGFtcGxlLmNvbS9zbS9tYWludGVuYW5jZQ/submodel-
elements/Reset/operation-status/4a2a7aaf-6f44-4f88-bba2-0b0d2d3e7a10
```

VIEW

VIEW is not bound to a dedicated HTTP method. In HTTP/REST APIs it is usually enforced on GET operations that only expose existence information, references, identifiers, or descriptor entries.

Example 1: retrieve references instead of full resource content.

```
GET /api/v3/submodels/aHR0cHM6Ly9leGFtcGxlLmNvbS9zbS9uYW1lcGxhdGU/$reference
```

Example 2: retrieve submodel references of an AAS.

```
GET /api/v3/shells/aHR0cHM6Ly9leGFtcGxlLmNvbS9hYXMvMTIzNA/submodel-refs
```

Example 3: retrieve descriptor entries without accessing the hosted AAS content itself.

```
GET /api/v3/shell-descriptors
```

Notes for access rule authors

- ROUTE literals SHOULD be defined against stable path families such as /shells/, /submodels/, /shell-descriptors/ or /lookup/ if authorization is intended for complete interfaces.
- Exact ROUTE literals such as /submodels/aHR0cHM6Ly9leGFtcGxlLmNvbS9zbS9tYWludGVuYW5jZQ/submodel-elements/Reset/invoke are useful for single sensitive operations.

Change Log

Changes w.r.t. V3.1 vs. V3.0.2

Major Changes:

- changed: <ReferenceAttribute> with more detailed definition in BNF Grammar and JSON Schema
- changed: <IdentifiableObject>, <ReferableObject>, <FragmentObject>, <DescriptorObject> with more detailed definition in BNF Grammar and JSON Schema
- changed: DateTime and Time literals in access rules now use explicit patterns based on AAS core meta MatchesXsDateTime and MatchesXsTime, which correspond to the XSD-based data types defined by the IDTA 01001 Part 1 Metamodel and are based on XML Schema xs:dateTime/xs:time/ISO 8601 lexical forms. Values require seconds, may include fractional seconds and timezone offsets, Time values without timezone information are interpreted as UTC, and incomplete values such as 09:00 need to be completed before serialization, e.g. 09:00:00Z. Calendar-specific invalid dates still need semantic validation beyond the lexical day, month and year ranges. [#63](https://github.com/admin-shell-io/aas-specs-security/issues/63)
- added: Query Filter and FragmentFieldIdentifiers (https://github.com/admin-shell-io/aas-specs-api/issues/517)
- added: new section in the annex on access control management and data authentification for implementing the digital product passport (DPP)
- added: Added chapter requirements.adoc and annex/requirements_mapping.adoc
- changed: Avoided provisional verbs in introduction.adoc, essential provision text has been moved to acccess-rule-model.adoc

Minor Changes:

- update: Revised description text to replace 'may include signing' / 'will be' with 'is signed' / 'be signed' for clarity and consistency.
- added: Support for supplementalSemanticIds in Access Rule FieldIdentifiers and FragmentFieldIdentifiers, including BNF, JSON Schema, and examples [#77](https://github.com/admin-shell-io/aas-specs-security/pull/77)
- added: Access control examples for REFERENCE & along with rules for creating only specific models, defined using BNF and JSON.
- added: Clarified <ReferenceAttribute> semantics and added a state-dependent filtering example where maintenance documents are filtered using machine state read via REFERENCE from another Submodel.
- added: Clarified ROUTE semantics for AAS HTTP/REST path matching and added informative ROUTE and request examples [#58](https://github.com/admin-shell-io/aas-specs-security/issues/58)
- added: Expanded the explanation of RIGHTS with a mapping to operation verbs and HTTP methods, including an illustrative table and diagram [#56](https://github.com/admin-shell-io/aas-specs-security/issues/56)
- moved: Moved and revised protection goals from introduction to own chapter
- updated: Editorial changes to the explanation of the access rule BNF
- added: Clarification of semver usage

Bugfixes:

- fix: Fixed direct FieldIdentifier operands in typed hex, bool, DateTime and time comparisons in BNF and JSON Schema, including implicit conversion semantics.
- fix: Disallowed direct FieldIdentifier-to-FieldIdentifier comparisons in BNF and JSON Schema. Comparing two fields requires an explicit cast around at least one operand.
- fix: Aligned date/time extraction functions ($dayOfWeek, $dayOfMonth, $month, $year) in JSON Schema to accept all dateTimeOperand types per BNF grammar.
- fix: Aligned comparison and cast operands in JSON Schema with BNF semantics, including boolean equality-only comparisons, ordered comparisons without boolean operands, restricted $dateTimeCast/$timeCast inputs, and no raw $boolean inside $match.

- changed: Fixed incorrect Fragment Field Identifiers in examples [#44 of Security Spec](https://github.com/admin-shell-io/aas-specs-security/issues/44)
- fix: access rule BNF

Changes w.r.t. V3.0.2 vs. V3.0.1

Bugfixes:

- removed: TREE
- fix: error in datetime BNF ObjectGroup
- fix: naming inconsistencies inside BNF and json schema (https://github.com/admin-shell-io/aas-specs-api/issues/547)
- fix: aligned BNF and json schema to api spec (https://github.com/admin-shell-io/aas-specs-api/issues/547)
- fix: BNF did not compile (https://github.com/admin-shell-io/aas-specs-security/issues/62)
- fix: BNF used incorrectly plural for USEACL & USEFORMULA (https://github.com/admin-shell-io/aas-specs-security/issues/61)
- fix: Regex FieldIdentifier was incorrect (https://github.com/admin-shell-io/aas-specs-security/issues/60)
- fix: ncorrect FieldIdentifier AAS Submodel References (https://github.com/admin-shell-io/aas-specs-security/issues/59)

Changes w.r.t. V3.0.1 vs. V3.0

Bugfixes:

- changed: Removed incorrect but required whitespaces from grammar and examples [#477 of API](https://github.com/admin-shell-io/aas-specs-api/issues/477)
- changed: fixed idShortPath definition in the BNF Grammar for the Query Language [#34](https://github.com/admin-shell-io/aas-specs-security/issues/34)
- changed: text serialization of Reference does not follow text serialization of Part 1 of References, the updated grammar will be added in the V3.1 [#33](https://github.com/admin-shell-io/aas-specs-security/issues/33)

Minor Changes:

- changed: correct example in Annex
- removed: remove <FieldIdentifierString> in grammar and use <FieldIdentifier> directly

Changes V3.0

This is the first release

Bibliography

[1] IDTA 01001 Specification of the Asset Administration Shell. Part 1: Metamodel, Version 3.2. Industrial Digital Twin Association (IDTA). Online. Available: https://industrialdigitaltwin.org/en/content-hub/aasspecifications

[2] IDTA 01002 Specification of the Asset Administration Shell. Part 2: Application Programming Interfaces, Version 3.2. Industrial Digital Twin Association (IDTA). Online. Available: https://industrialdigitaltwin.org/en/content-hub/aasspecifications

[3] IDTA 01003-a Specification of the Asset Administration Shell. Part 3a: Data Specification – IEC 61360, Version 3.1. Industrial Digital Twin Association (IDTA), January 2025. Online. Available: https://industrialdigitaltwin.io/aas-specifications/IDTA-01003-a/v3.1/index.html

[4] IDTA-01005 Specification of the Asset Administration Shell. Part 5: Package File Format, Version 3.1. Industrial Digital Twin Association (IDTA), January 2025. Online. Available: https://industrialdigitaltwin.io/aas-specifications/IDTA-01005/v3.1/index.html

[5] Tom Preston-Werner. Semantic Versioning. Version 2.0.0. Online. Available: https://semver.org/spec/v2.0.0.html

[6] Secure Download Service, Plattform Industrie 4.0, October 2020. Online. Available: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/secure_downloadservice.pdf

[7] Dataspace Protocol 2024-01. Edited by Sebastian Steinbuß, Julia Pampus, James Marino. International Dataspaces Association. Online. Available: https://docs.internationaldataspaces.org/ids-knowledgebase/v/dataspace-protocol

[8] Vincent Hu, David Ferraiolo, Rick Kuhn, Adam Schnitzer, Kenneth Sandlin, Robert Miller and Karen Scarfone, “Guide to Attribute Based Access Control (ABAC) Definition and Considerations”, NIST Special Publication 800-162, Jan. 2014. [Online]. Available: http://dx.doi.org/10.6028/NIST.SP.800-162

[9] “eXtensible Access Control Markup Language (XACML)”, OASIS Standard, Version 3.0, 22. Jan. 2013, [Online]. Available: http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.pdf

[10] ITU-T Recommendation X.509 (2005), ISO/IEC 9594-8:2005, Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks.

[11] Decentralized Identifiers (DIDs) v1.0, Core architecture, data model, and representations; https://www.w3.org/TR/2022/REC-did-core-20220719/; W3C; July 19, 2022

[12] IEEE Standard for Local and Metropolitan Area Networks—Secure Device Identity. IEEE Std 802.1AR-2018, August 2018.