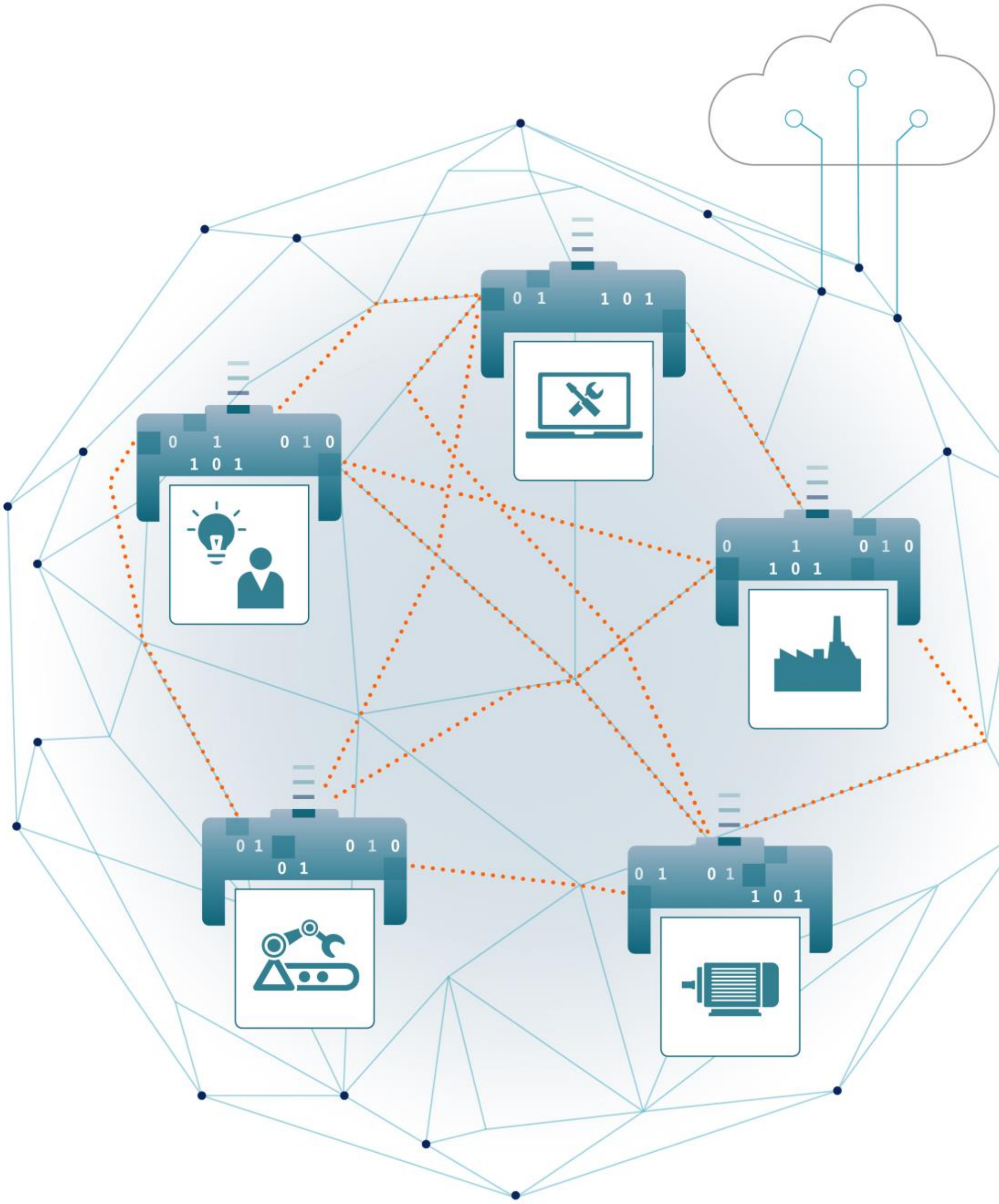
PLATTFORM INDUSTRIE 4.0

Discussion Paper



Functional View of the Asset Administration Shell in an Industrie 4.0 System Environment

Imprint

Publisher

Plattform Industrie 4.0
Bülowstraße 78
10783 Berlin
geschaeftsstelle@plattform-i40.de

Text and Editing, Design and Picture Credits

Plattform Industrie 4.0
Bülowstraße 78
10783 Berlin

Status

April 2021

This publication is issued by Plattform Industrie 4.0. The publication is distributed free of charge and is not intended for sale.

Contents

1. Terms and Definitions

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

access control

protection of system resources against unauthorized access; a process by which use of system resources is regulated according to a security policy and is permitted by only authorized entities (users, programs, processes, or other systems) according to that policy

SOURCE:

- IEC TS 62443-1-1:2009]
- Glossary Industrie 4.0

Asset Administration Shell (AAS)

standardized digital representation of an asset.

- Note 1: Asset Administration Shell and Administration Shell are used synonymously.
- Note 2: Each administration shell can contain one or multiple submodels
- Note 3: The administration shell can be passive, re-active, or pro-active
- Note 4: The administration shell exists within one phase or across different phases of the lifecycle.
- Note 5: Assets are part of an I4.0 component in an I4.0 system

SOURCE:
Glossary Industrie 4.0 (work in progress)

AAS infrastructure

subset of I4.0 infrastructure services to create, register and search Asset Administration Shells and corresponding endpoints.

application

software functional element specific to the solution of a problem in industrial-process measurement and control

- Note 1 to entry: An application can be distributed among resources and may communicate with other applications.
- Note 2 to entry: This definition is taken from IEC TR 62390:2005-01, section 3.1.2. In this specification, the problem space is extended to the whole application domain of Industrie 4.0.

SOURCE:
IEC TR 62390:2005-01, section 3.1.2.

application relevant service

software service that implements application-specific functionality and makes use of infrastructure services.

application component

software component that makes use of infrastructure services, sub models and asset related services to implement application specific functionality.

- Note 1 to entry: An application component can be any kind of software service (e.g. any other IT application, an AAS, a sub model service or an asset related service.)

asset

entity which is owned by or under the custodial duties of an organization, having either a perceived or actual value to the organization.

SOURCE:

- IEC TS 62443-1-1:2009, 3.2.6 (modified)
- ISO/IEC 20924:2018 Information technology – Internet of Things (IoT) – Vocabulary (modified) [Glossary Industrie 4.0]

asset related service

application-relevant software service that offers asset related functionality available within an I4.0 System Environment. Asset related services are managed by infrastructure services and accessed via AAS.

- Note 1 to entry: An asset related service may be implemented according to any given specification.
- Note 2 to entry: In Industrie 4.0 asset related services are described or implemented by submodels as defined in [6].
- Note 3 to entry: An asset related service may be provided by the asset itself or any system able to provide information or functionality for the asset
- Note 4 to entry: In [1] "asset related services" are named "asset services". Both terms are used equivalent.

computing infrastructure

computational foundation of a software application from the perspective of this application. It consists of the two elements computing capacity and infrastructure services.

endpoint

basic address under which an interface of a software service is accessible.

Note 1 to entry: Dependent on the implementation technology this basic address might be enhanced by applications by further details to address the complete set of operations offered by the interface of the software service.

infrastructure service

software service that is used by different application relevant services or applications in the same way. e.g. mediate, enable and support the interaction with and between I4.0 Components

Note 1 to entry: In this sense, the service is considered to be systemic relevant.

interaction

behavior that is specified by a sequence of messages between two or more system components, each of them comprising primitives related to call events, and/or notifications related to signal events

Note 1 to entry: In an extreme case, an interaction just consists of one primitive or one notification.

SOURCE:

- DIN SPEC 16593-1 [3] (modified)
- Glossary Industrie 4.0

interaction domain

set of interacting components defined by its constituting components or by other criteria (e.g. geographical or time-based filters)

SOURCE:

- DIN SPEC 16593-1 [3] (modified)
- Glossary Industrie 4.0

interaction pattern

typical sequence of messages (primitives or notifications) between interacting components as a solution to often-posed interaction needs

Note 1 to entry: Usually, in an interaction pattern the entities follow pre-defined roles (e.g. consumer/provider, publisher/subscriber) and associated constraints (e.g., order and direction of primitives/notifications).

SOURCE:

- DIN SPEC 16593-1 [3] (modified)
- Glossary Industrie 4.0

interaction policy

textual or formal specification of constraints or conditions on the interactions between components that are valid in a specified interaction domain

SOURCE:

- DIN SPEC 16593-1 [3] (modified)
- Glossary Industrie 4.0

interface

defined connection point of a functional unit which can be accessed by other functional units.

- Note 1 to entry: “Defined” means that the requirements and the assured properties of this connection point are described.
- Note 2 to entry: The connection between the interfaces of function units is also called an interface.
- Note 3 to entry: In an information system, the defined exchange of information takes place at this point.
- Note 4 to entry: Interface places certain requirements on the connection that is to be made.
- Note 5 to entry: Interface demands certain features.

SOURCE:

- Glossary Industrie 4.0
- DUDEN (modified)
- ISO/IEC 13066-1:2011(en), 2.15 (modified)
- DIN EN 60870-5-6:2009-11 (modified)
- DIN IEC 60625-1:1981-05 (modified)

I4.0 application

application dedicated to serve the vision and objectives of Industrie 4.0.

- Note 1 to entry: Synonym to application in [12], i.e., “software functional element specific to the solution of a problem in industrial-process measurement and control.”
- Note 2 to entry: Software components of an I4.0 application (application components) are users of the I4.0 infrastructure services.

I4.0 component

globally uniquely identifiable participant with communication capability consisting of administration shell and asset within an I4.0 system which there offers services with defined quality of service characteristics

- Note 1 to entry: For its services and data, the I4.0 component offers protection appropriate with the task.
- Note 2 to entry: An I4.0 component can represent a production system, a single machine or station, an assembly within a machine, or even an automation device.

SOURCE:

- Glossary Industrie 4.0 (modified)

I4.0 infrastructure

synonym to I4.0 platform

I4.0 platform

distributed software platform that enables the interaction between I4.0 components supporting the composition types of both collaboration and cooperation by means of dedicated infrastructure services.

- Note 1 to entry: This definition interprets the definition of an I4.0 platform of [7] to the context of this specification.

SOURCE:

- DIN SPEC 16593-1 [3]

I4.0 system

Set of interacting I4.0 components

- Note 1 to entry: A system may be present as a component in a further I4.0 system.
- Note 2 to entry: Flexibility, transformability, etc. are features of an I4.0 system.
- Note 3 to entry: This definition is more stringent refinement of [7].

operation

specification or realization of a behavior of a component in terms of procedures

- Note 1: The term method is synonym to operation in the IT domain
- Note 2: an operation has a name and a list of parameters [ISO 19119:2005, 4.1.3]
- Note 3 to entry: software services implement interfaces by means of operations and events.

SOURCE:

- DIN SPEC 16593-1 [3]

policy

constraints or conditions on the use, deployment or description of an owned entity as defined by any participant.

- Note 1 to entry: This definition is taken from OASIS RM SOA [4] and DIN SPEC 16593-1 [3].

service

limited scope of functionality which is offered by an entity or organization via interfaces.

Note 1 to entry: This definition is taken from [7].

Note 2 to entry: See the conceptual discussion of the term service in section 0.

Software service

service that is offered by one or more software components.

Note 1 to entry: Software services implement interfaces by means of operations and events.

2. Symbols and abbreviated terms

AS	Administration shell
AAS	Asset administration shell
API	Application programming interface
I4.0	Industrie 4.0 or Industry 4.0
IAM	Identity and Access Management
IDS	International Data Spaces (IDS)
IIC	Industrial Internet Consortium
IIoT	Industrial Internet of Things
IIRA	Industrial Internet Reference Architecture
IoT	Internet of Things
RAMI4.0	Reference Architecture Model Industrie 4.0
REST	REpresentational State Transfer
SM	Sub-model (of an AAS)
UML	Unified Modeling Language

3. Foreword

One of the key concepts of the initiative Industrie 4.0 is the Asset Administration Shell (AAS). First implementations of the AAS and mappings to various technologies are available. Nevertheless, a comprehensive picture from a functional point of view that also includes the whole system environment is still missing.

On the one hand, this is necessary such that a broad community may better understand the objectives of this concept and is therefore able to generate benefits from its usage. On the other hand, it is fundamentally and systematically required to analyze and conceptualize non-functional requirements such as IT security and privacy, following the "security by design" and "privacy by design" principles.

This document was developed from December 2019 to April 2021 by sub working group "Infrastructure of the Asset Administration Shell" and reviewed by sub working group "Asset Administration Shell" of Platform Industrie 4.0 working group "Reference Architectures, Standards and Norms".

It extends the Usage View of an Administration Shell document with a corresponding Functional View.

The Editors would like to thank all participants in the discussions to develop and review this document.

4. References

[1] Plattform Industrie 4.0: Usage View of the Asset Administration Shell. Discussion Paper. https://www.plattform-i40.de/PI4.0/Redaktion/DE/Downloads/Publikation/2019-usage-view-asset-administration-shell.html

[2] Industrial Internet Consortium (IIC): The Industrial Internet of Things Volume G1: Reference Architecture, Version 1.9, June 19, 2019.

[3] DIN SPEC 16593-1: Reference Model for Industrie 4.0 Service Architectures - Part 1: Basic concepts of an Interaction-based Architecture (IBA). https://www.beuth.de/de/technische-regel/din-spec-16593-1/287632675

[4] OASIS, Reference Model for Service Oriented Architecture 1.0, Committee Specification 1, 2006. http://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf

[5] VWS in der Praxis: BMWi (Ed.). (2019). Diskussionspapier Verwaltungsschale in der Praxis—Wie definiere ich Teilmodelle, beispielhafte Teilmodelle und Interaktion zwischen Verwaltungsschalen (Version 1.0) in. https://www.plattform-i40.de/PI4.0/Redaktion/DE/Downloads/Publikation/2019-verwaltungsschale-in-der-praxis.pdf?__blob=publicationFile&v=7

[6] AAS in Detail Part 1: Details of the Asset Administration Shell - Part 1 - The exchange of information between partners in the value chain of Industrie 4.0 (Version 2.0.1), Plattform Industrie 4.0 und ZVEI, November 2019.

https://www.plattform-i40.de/PI4.0/Redaktion/DE/Downloads/Publikation/Details-of-the-Asset-Administration-Shell-Part1.html

[7] VDI (Ed.). (2019). Industrie 4.0 Begriffe/Terms—VDI-Statusreport April 2019. Beuth. https://www.vdi.de/ueber-uns/presse/publikationen/details?tx_vdipublications_publication-details%5Bpublication%5D=267&cHash=8af6e9a628a05abe41337bbadf71b9b4 [p. 13])

[8] BITKOM (Ed.): Vorschlag zur systematischen Klassifikation von Interaktionen in Industrie 4.0 Systemen. Platform Industrie 4.0 and BITKOM White Paper (2020). https://www.plattform-i40.de/PI40/Redaktion/DE/Downloads/Publikation/Bitkom_Thesenpapier.html

[9] VDI/VDE 2193 Blatt 1:2020-04 Language for I4.0 Components - Structure of messages. Beuth Verlag (2020). https://www.beuth.de/en/technical-rule/vdi-vde-2193-blatt-1/318387425

[10] VDI/VDE 2193 Blatt 2:2020-01 Language for I4.0 components - Interaction protocol for bidding procedures. Beuth Verlag (2020). https://www.beuth.de/en/technical-rule/vdi-vde-2193-blatt-2/314114399

[11] International data Spaces Association (Ed. Sebastian Steinbuss): Usage Control in the International Data Spaces. Position Paper, version 2.0, November 2019. https://www.internationaldataspaces.org/wp-content/uploads/2019/11/Usage-Control-in-IDS-V2.0_final.pdf

[12] IEC CD 63278-1 ED. Asset administration shell for industrial applications – Part 1: Administration shell structure. IEC 2020.

5. Introduction

This document aims at clarifying the concept, roles and functional areas of infrastructure services and application components in a system environment. It motivates and introduces the concept of an Industrie 4.0 Infrastructure as part of an Industrie 4.0 System Environment and its enabling functional areas to support interactions between software components in an Industrie 4.0 system and in Industrie 4.0 Applications. These software components may be Industrie 4.0 Components, i.e., they comprise an Asset Administration Shell (AAS) and an associated asset or be application components. This document is part of a series that follows the viewpoint approach for analyzing and designing IIoT systems as described by the Industrial Internet Reference Architecture (IIRA) of the Industrial Internet Consortium (IIC) [2].

The IIRA identifies the relevant stakeholders of IIoT systems (such as Industrie 4.0 systems) and determines the proper framing of concerns, i.e., any topics of interest pertaining to a system. The four IIRA viewpoints are:

- Business Viewpoint
- Usage Viewpoint
- Functional Viewpoint
- Implementation Viewpoint

As illustrated in Figure 1, this document defines the IIRA Functional View of an Industrie 4.0 system environment and as such closes the gap between the Usage View of the AAS [1] that describes the requirements from an application point of view [1], and the Implementation View of the concept of an Asset Administration Shell (AAS) in Industrie 4.0.

> Note 1: According to the IIRA, an (architecture) view comprises the collection of ideas describing, analyzing, and resolving the set of specific concerns in a viewpoint using the conventions specified in that viewpoint.

In addition to specifying the concept of the Asset Administration Shell (AAS) there is a need to consider the characteristics of a whole system environment

- when designing industrial applications based upon I4.0 concepts,
- when trying to understand what is functionally determined by Industrie 4.0 and what not (boundaries of an Industrie 4.0 system),
- when trying to understand, what is part of an application, an I4.0 system and an I4.0 infrastructure and how to integrate application and infrastructure functionality into an I4.0 system,

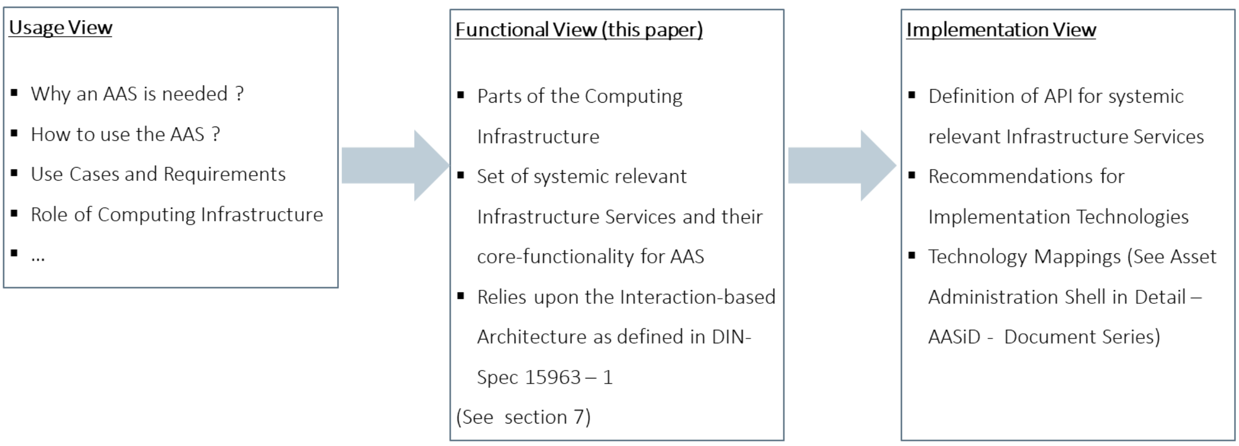


Figure 1: Linkage between the Usage, Functional and Implementation View

- when trying to specify the functions of a computing infrastructure for interacting AAS instances,
- when trying to follow “security by design” (e.g., according to IEC 62443) and “privacy by design” (to support the Industrie 4.0 vision of Autonomy/Sovereignty),
- when mapping I4.0 concepts and specifications to other initiatives (e.g., GAIA-X and International data Spaces), and, finally
- when standardizing I4.0 concepts.

Consequently, this document comprises the following steps:

1. It discusses the term computing infrastructure (see section 0), originally introduced in the document “Usage View of the AAS” [1], from a functional viewpoint. Furthermore, this term is associated with the concepts Industrie 4.0 system, domain and policy as introduced in the DIN SPEC 16593-1 [3].
2. It formulates the value proposition of the AAS to guide the derivation of the functional view of the AAS.
3. It provides an Industrie 4.0 Service Model (section 0) as well as an Industrie 4.0 System Model (section 0).
4. It derives and defines the set of infrastructure services in order to enable the interaction between as well as implementation and execution of Asset Administration Shells. Infrastructure Services provide core functions for which Industrie 4.0 defines an interface and a corresponding data model. Basically, they provide those functions which are deemed necessary for a computing infrastructure or are indispensable for an AAS implementation.

   Note: Later on, this specification may be used as basis for a series of further AAS Implementation View documents („AAS in Detail Series“).

Figure 2 positions this document “Functional View of an AAS” in the context of other Industrie 4.0 specifications and applies the four architectural viewpoints of IIRA mentioned above.

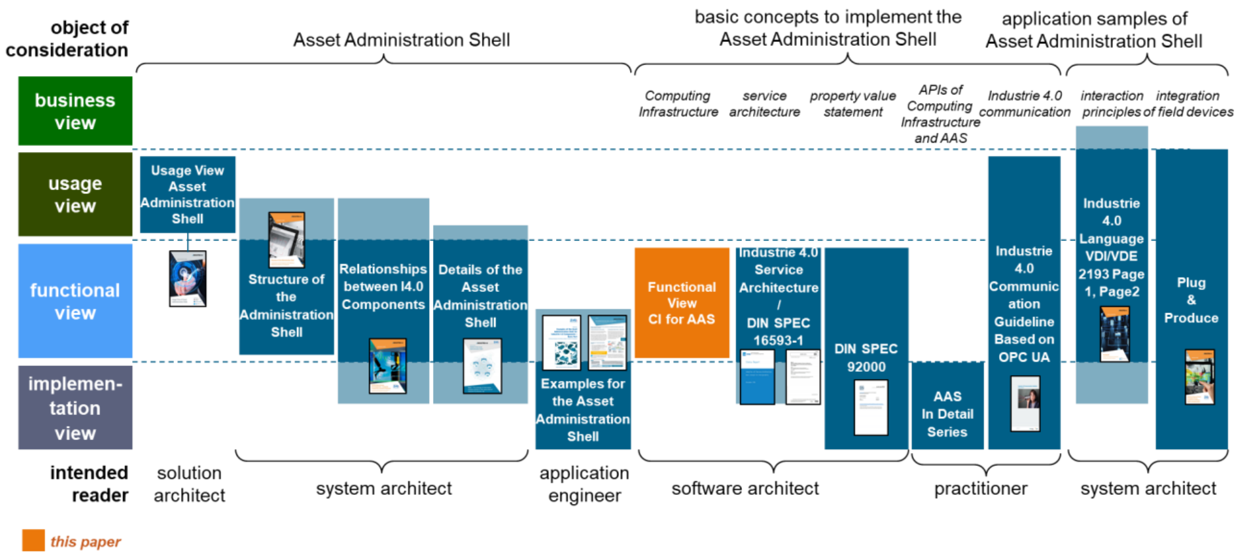


Figure 2 Positioning of the Functional View of an AAS in the context of other Industrie 4.0 Specifications

With the release of the Industrie 4.0 Strategy in 2012 concept discussions started about technical concepts to support the value proposition stated in the paper. On core concept comprised the virtualization of physical assets and resulted in the definition of the “Asset Administration Shell” – a specific foundation for the implementation for the industrial context of an abstract “Digital Twin” concept that already existed.

After defining the abstract concept of the Administration shell, the community members from different fields of expertise realized the challenge to specify the common denominator of these concepts for the variety of thematic problems to be solved in the industry domain. Therefore, it was decided complete the “top down” design approach by a “bottom-up” approach in which an information model of an asset administration shell was defined in the beginning of the technical specification.

This resulted in a tremendous improvement of a common understanding and also helped to form productive expertise groups around the definition of these elements. Still there was a gap between the abstract usage viewpoint on the administration shell and the existing specification of the asset administration shell in the documents AASiD Part 1 (information model) and Part 2 (API).

Being aware of both threads coming from top down and bottom up – the functional view is trying to build a bridge between these approaches – assuming that both are describing parts of the same Industrie 4.0 system.

The authors did not identify any elements in the one or the other document (usage view or AASiD) that are conflicting – it rather describes an implementation-neutral view on the functional aspects of the AAS to implement an Industrie 4.0 system where the specified elements specified in the AASiD documents are a valid implementation.

As the functional view itself cannot define – if the AASiD documents are an exhaustive specification for the functional definition, the authors tried to relate some of the elements that are already well understood – but also leaves openness for discussion where that was not yet achieved.

The target of these papers is to provide a seamless set of architecture viewpoints to trace between technical implementation and business value of an AAS.

6. Motivation

Several Industrie 4.0 documents, e.g., [5], [6] and [12], describe the structure and content (meta-)model of an AAS. All of them address several aspects of AAS functionality and its computing infrastructure driven by different application scenarios. Examples for those scenarios but not limited to are structural aspects, service aspects, content aspects, communication aspects, technology aspects.

The AAS Usage View document [1] sets a frame for a general understanding of the value proposition of the AAS (see section 0) from a usage point of view. The following main statements are repeated in the subsections in order to motivate the derivation of the required functions of an AAS infrastructure and system environment when implementing the AAS concept.

> Note: In the following, the term “asset related service” is used according to [1] as a uniform conceptual requirement to access information and functions of an asset.

6.1 Asset

In [1], an asset is characterized as follows:

- Item which has a value for an organization, and which is administrated individually for this reason
- An asset belongs to an owner, the owner may change over the lifecycle of an asset
- During a transfer of ownership various information about the asset may be handed over: the new owner is responsible to manage the registration of the functions made available by the asset (in [1] called asset related services) and to integrate this information suitably
- An asset may comprise own computing resources and implementations of specific asset related services: it is in the responsibility of the owner of the asset to integrate the computing resources and implementations suitably into the computer infrastructure
- Relevant relations between assets (from the perspective of a stakeholder w.r.t. the asset) have counterparts in form of relations between the corresponding asset related service registries

6.2 Stakeholder

In [1], the stakeholders with respect to an asset are as follows:

- Organization that has an interest in the asset under consideration, therefore the stakeholder creates an asset related service registry and possibly in addition
  - Asset related services (and then being the owner of these asset related services)
  - software applications (and then being the owner of these software applications)

- The owner of an asset is a specific stakeholder w.r.t. an asset
  - The owner defines the access and usage policy of any asset related service provided by the asset itself, i.e. asset related services running on the computing resources of an asset
  - Any other stakeholder w.r.t the asset may create own asset related services (and consequently defines the access and usage policy of these asset related services)

6.3 Asset Administration Shell

Based upon [1], Figure 3 describes a high-level exemplary overview of the AAS concept and its usage with the following main characteristics:

- The AAS comprises an asset related service registry of an asset from the perspective of an organization having an interest in the asset. Such an asset related service registry declares all asset related services of the asset, which are of interest for the organization.
- The AAS provides secure access to asset related services.
- Asset related services may be standardized by standardization organizations.
- The AAS may offer structuring possibilities, including encapsulation, following the relationships between assets.

The AAS concept does not prescribe a given way of deployment and implementation. It is dependent of the asset type and its capabilities.

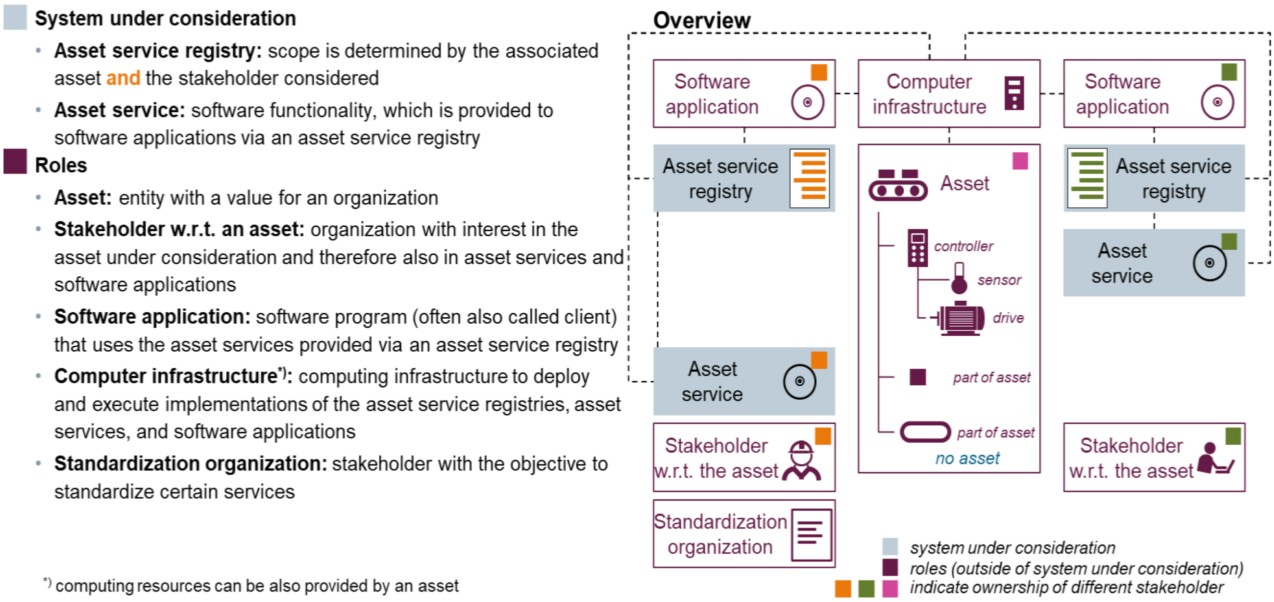


Figure 3 System under Consideration in the Usage View of an AAS [1]

7. Value proposition of the Asset Administration Shell

To explain functionality of infrastructure services it is a necessary pre-condition to define the value proposition of the AAS.

This value proposition guides the necessary decisions on functionality of infrastructure services and lays the foundation for further design decisions e.g.

- How to protect the IP of stakeholders?
- How to integrate the installed base?
- How to systematically integrate application-relevant functionality into an I4.0 System Environment?
- How to support the interactions with and between I4.0 components by means of Industrie 4.0 infrastructure services?

To address such questions the AAS value proposition comprises of the following three points:

1. The AAS defines a unifying concept to structure and access asset related services, which is a value by its own from the perspective of stakeholders, owners and users. It eases e.g.
    - Tender development and tender preparation processes
    - Procurement processes
    - Life Cycle Management processes
    - Engineering processes
    - Quality Assurance processes

2. The AAS defines a unifying concept for information structuring by transferring the component-based view for software systems from the IT world to an asset-oriented view in I4.0 Systems. It is an application of the long-standing “information hiding” principle in software engineering.

The transfer of the component-based view from IT to an asset-oriented view in I4.0 is of value because:

- It is an easy to learn concept applicable to all assets and by all stakeholders on all system levels in the same way.
- It builds a foundation to ease system integration and to reduce engineering effort in even heterogeneous systems.
- It builds the foundation for plug & play in order to improve the flexibility of I4.0 Systems.
- It lays the foundation to integrate the installed base by making existing products I4.0 Assets without a need to change them (also called brown field integration) In this sense, it shows a way towards I4.0 for all market participants and provides protection of investment.

3. The AAS defines a unifying concept for secure and reliable restriction of use to AAS as well as to asset related services.

This is of value because:

- It is an important concept for system integrity based on trusted assets.
- It is an important concept for system protection against cyber-attacks.
- It is an important concept to combine access control with usage control e.g., license policies as a basis for commercialization of Assets and Asset related services.
- It is an important concept for definition of ownerships and therefore the basic concept for the IP protection of all stakeholders w.r.t. an asset.

These three points are not totally new. In the figurative sense they are the translation of crucial success factors of the communication industry to Industrie 4.0.

Two foundations valid for every communication protocol stack enable communication industry to integrate and combine different technologies and products of different vendors and to maintain communication systems over long periods.

- Both Systems are guided by a reference model for system partitioning on a functional level.
- Both Systems use unifying access points to enable access to services.
- Both Systems use unified Interaction models to exchange information.

In this sense the AAS can be seen as the "Service Access Point" for asset related services of an Asset.

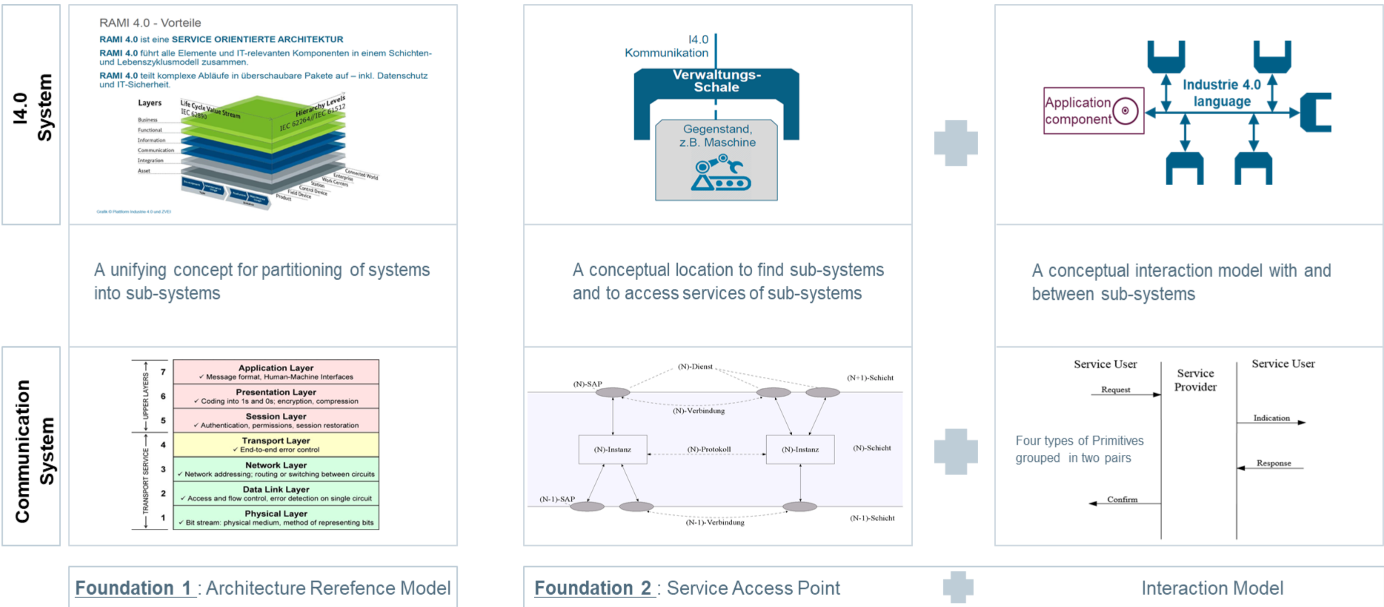


Figure 4 Similarity of Communication Systems and I4.0 Systems

8. Computing Infrastructure

The functions of the computing infrastructure lay the foundation for the creation, deployment, operation, maintenance, usage, and deletion of Asset Administration Shells and support the interaction between them. There are two independent concepts within a computing infrastructure which constitute the necessary functionality when combined with each other:

- computing capacity and
- software services.

8.1 Computing Capacity

Computing capacity is represented by a network of computing nodes (hardware devices including firmware and operating system and basic communication services) which can communicate with each other and offer functionality for computation and communication to the software services. In practice there is a big variety of configurations of those networks.

Industrial IoT Networks are examples for the provision of computing capacity. These networks apply various IT deployment concepts such as

- Cloud Computing
- Fog Computing
- Edge Computing
- Desktop Computing
- Embedded Computing
- Special computing e.g., real-time computing in control systems.

Industrial IoT Networks are examples for computing capacity, some more are illustrated in Figure 5.

The offering of different deployment scenarios by different architectures of computing capacity is of high value in Industrie 4.0 implementations. It enables to fulfill various quality requirements and to adapt software applications to business needs. Clause 12 will describe examples for different deployments of Asset Administration Shells to different architectures of computing capacity.

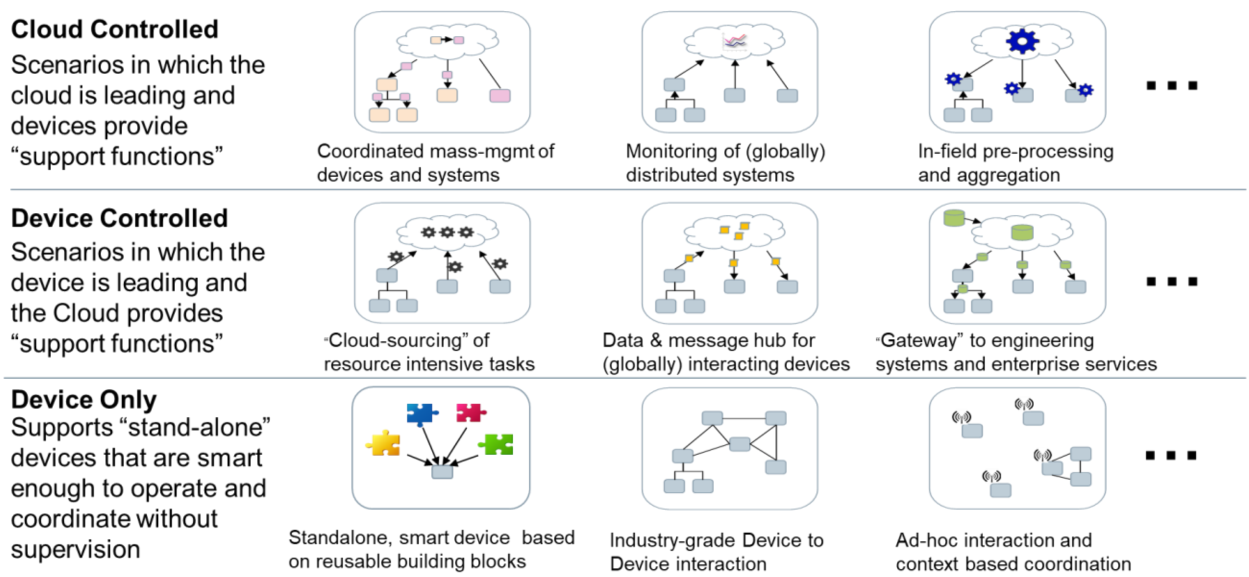


Figure 5 Examples of computing capacity

8.2 Software Services

Software services are functions of a computing system that are offered by software components. They provide functionality by making use of the computing capacity. In this sense a software service is a basic concept for application design whereas the computing capacity is the basic concept for application deployment and operation.

> Note: The term service used here in its most generic sense as defined in the Industrie 4.0 Glossary as "demarcated scope of functionality which is offered by an entity or organization via interfaces".

In practice this leads to a „decoupling of design from deployment of applications" which is supported by many technology trends. Best known and probably most popular is the Internet as a set of interconnected computing nodes which provides the computing capacity for World Wide Web applications.

Software services themselves split into the following two main groups as illustrated in Figure 6:

1) "Systemic relevant software services are characterized by the fact that they are used by application relevant services in the same way. This means that they must have a uniform syntax and a uniform semantic for all application relevant services. They are named "Infrastructure Services".

2) "Application relevant software services" are characterized by the fact that they make use of infrastructure services and use them for their own functionality in a particular application. In I4.0 Applications they are named "Application Components" as set forth in clause 9.2"

   A subset of application components are asset related services. They are characterized by the fact that they are made available by an AAS. In I4.0 Systems this is always the case because those systems consist of interacting I4.0 components. Asset related services are managed by infrastructure services and in addition make use of them as any other application component does.

   Examples for asset related services may be calibration services for a CNC machine or manipulation services for robot arms.

   > Note: Asset related service does not mean that the service is deployed onto the asset. It means that the service is somehow related to the asset and therefore is made available by its AAS as single-entry point to it.

The document will focus on infrastructure services for I4.0. It will illustrate why they are needed and what are the relationships to application components to I4.0 Application Components.

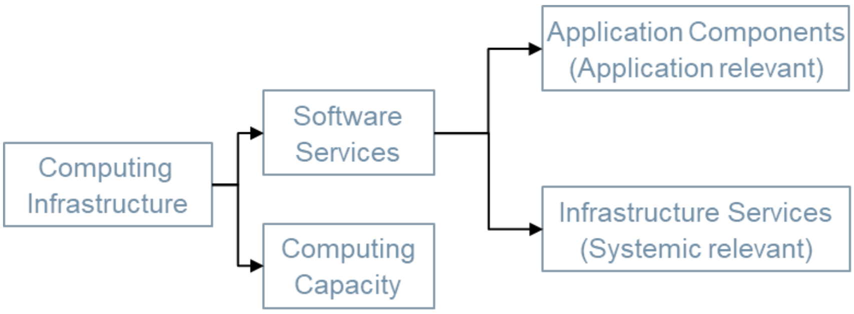


Figure 6 General Elements of Computing Infrastructure

9. Basic Models

9.1 Industrie 4.0 Service Model

The Industrie 4.0 Service Model is derived from the Interaction-based Architecture as specified in the DIN SPEC 16593-1 [3]. The Interaction-based Architecture clarifies the relationships between the I4.0 components covering all three AAS interaction types (passive, re-active, pro-active) explained above. Basically, it distinguishes two ways of how the behavior between (I4.0) components expressed as a sequence of interactions may be specified:

1) by means of procedures (here: operations) or
2) by means of state machines.

In this document, the focus lies upon the first type of interaction behaviors, i.e. procedure-based interactions. State-machine based interactions refer to pro-active AAS interactions types and will be discussed in a later version of this document.

The Industrie 4.0 Service Model is illustrated in Figure 7.

It may be mapped to both procedure-based and state-machine based interactions. It distinguishes between associated concepts on several levels (from left to right):

- technology-neutral level: concepts that are independent from selected technologies which could follow an architectural style (e.g., client/server or publish/subscribe)
- technology-specific level: concepts that are instantiated for a given technology (e.g., https, 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 Industrie 4.0 system.

The concepts that are dealt with in this AAS Functional View document are typically those of the technology-neutral level. However, in order to avoid terminological and conceptual misunderstandings, the whole Industrie 4.0 service model is provided here.

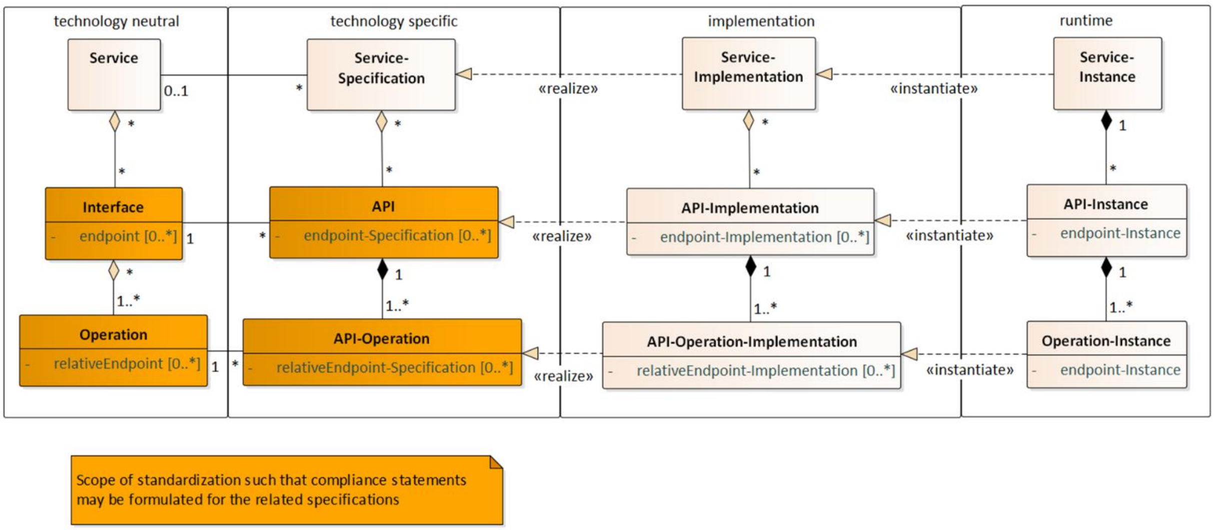


Figure 7 Industrie 4.0 Service Model

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. Examples are the infrastructure services as described and categorized into abstract service categories in this document (see section 10). However, also the application-relevant asset related services belong to this level.
- 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 being mapped to application programming interfaces (API) in the technology-specific level (see below). One interface may be mapped to several APIs depending on the technology and architectural style being used, whereby these API mappings also have to be standardized for the sake of interoperability.
- Interface-Operation: An interface is specified by means of operations and their interrelationships. For more details see the interaction policies and patterns defined in DIN SPEC 16593-1 [3].

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. These may be I4.0-defined standard APIs but also other, proprietary APIs.

  Note: Such a technology-specific service specification may but does not need to be derived from the service description 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 to be supported.
- API (Application programming Interface): Specification of the set of operations and events that form 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 may be called through an API. It is derived from the Interface-Operation description on the technology-neutral level. Hence, if there are several selected technologies, one interface-operation may be mapped to several API-operations.

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 the communication. Additional it has an identifier to be identifiable within a given context.
- API-Instance: instance of an API-Implementation which has an endpoint to get the information about this instance and the related operations.
- Operation-Instance: instance of an API-Operation-Implementation which has an endpoint to get invoked.

Note 1: For the sake of simplicity, this I4.0 Service Model does not illustrate asynchronous "notifications" (exposed as signal events) as proposed in [3]. For this consideration, they are considered as "operations" (exposed as call events), accordingly.

One important take-away 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.

It is important to understand that the Industrie 4.0 Service Meta-Model is applied to both asset related services and infrastructure services as introduced and distinguished above in section 0. Its instantiation for infrastructure services is the core of this AAS Functional View specification illustrated in Figure 8.

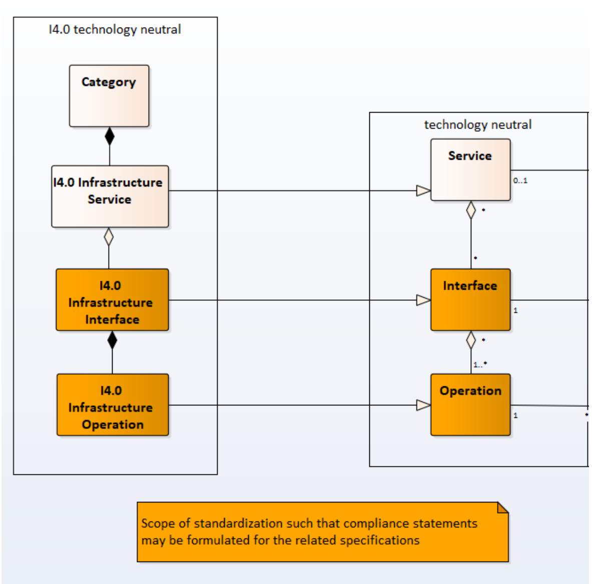


Figure 8 Service Model mapped to I4.0 Infrastructure Services

As highlighted above the design and specification of services, if asset related services or infrastructure services, is typically outside the scope of an Industrie 4.0 specification. It is the interface that is to be specified as reusable unit and to be mapped to the AAS meta-model concepts. An interface consists of Interface-Operations (to call procedures within software components) and notifications (to receive asynchronous events from software components).

If the Interfaces should be modeled with elements specified in the AAS meta-model, the submodel elements "Operation" and "Event" could be used. The data elements shall be used to define parameters of both operations and notifications.

Note that this mapping is outside the scope of Industrie 4.0 standardization and shall be carried out by a software architect resp. software engineer. Typically, multiple mapping options are possible depending on the architectural style defined in the technology-neutral level and used in a system architecture.

The following example is given in order to motivate the general understanding:

As part of an energy management interface for an asset (e.g., field device or a machine tool) an operation "SetEnergyLevel (level: int)" may be specified. This may be mapped either

- to one AAS submodel operation with a same or similar name that is functionally equivalent,
- to a transactional series of AAS submodel operations to machine tool sub-component, or
- to a "write" or "update" operation upon an AAS data element "energyLevel"

depending on the complexity of the asset and the implementation environment.

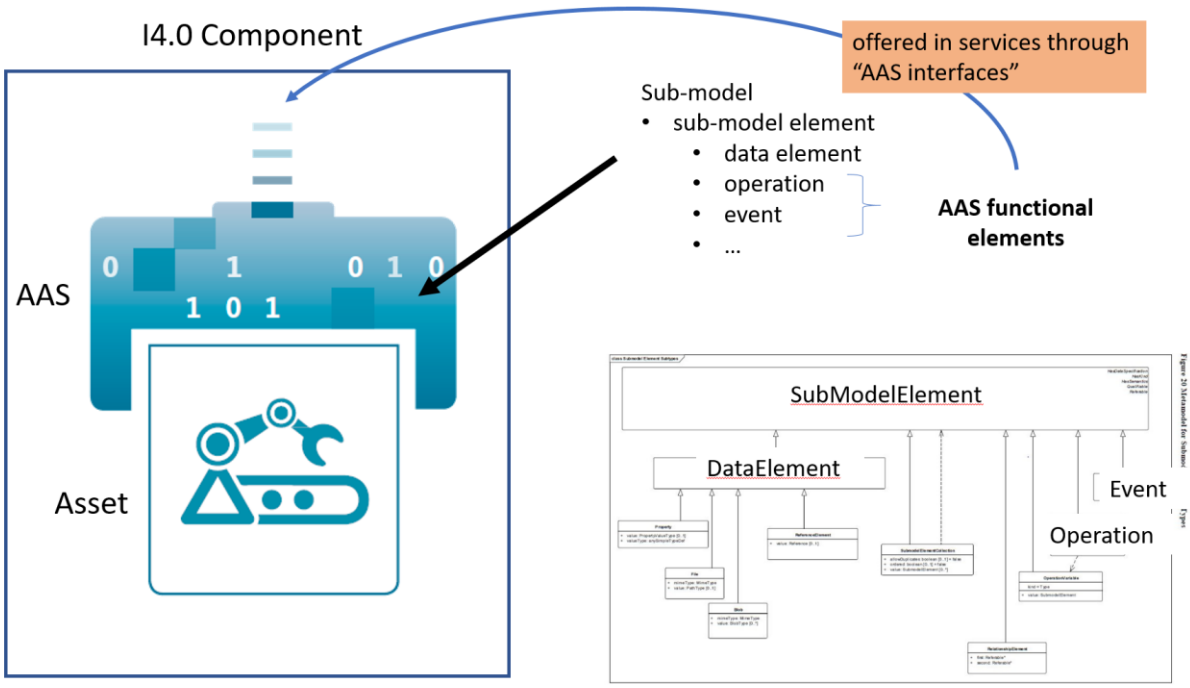


Figure 9 AAS Management Functions offered as Infrastructure Services

9.2 Industrie 4.0 System Model

9.2.1 Foundation and Compliance Mode 1

This document adopts the system-component model as defined in the DIN SPEC 16593-1 [3] and ISO/IEC 19746-2:2009. This means that Industrie 4.0 systems are built of interacting Industrie 4.0 components as illustrated in Figure 10.

The benefit is that the boundary of I4.0 Systems is clearly defined by the set of uniquely identifiable I4.0 Components.

All software components that do not comply with the definitions of an I4.0 Component, i.e.

- that do not follow the specification of the AAS meta-model when specifying their data model and functions, and
- that do not offer their data and functionality through an AAS interface,

are outside of an I4.0 System.

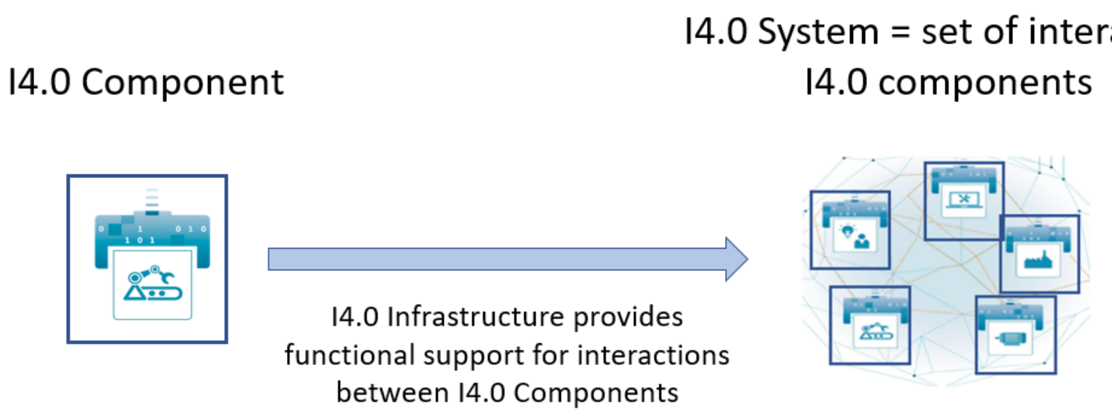


Figure 10 I4.0 System as a set of interacting I4.0 Components

They are part of two other functional areas: either part of an I4.0 Application, or part of an external infrastructure as illustrated in Figure 11.

All together they constitute the Industrie 4.0 System Environment that is realized by means of the capabilities of the underlying computing capacity. Application components within an Industrie 4.0 Application may be users (clients, consumers) of the Industrie 4.0 Components following their exposed AAS interfaces.

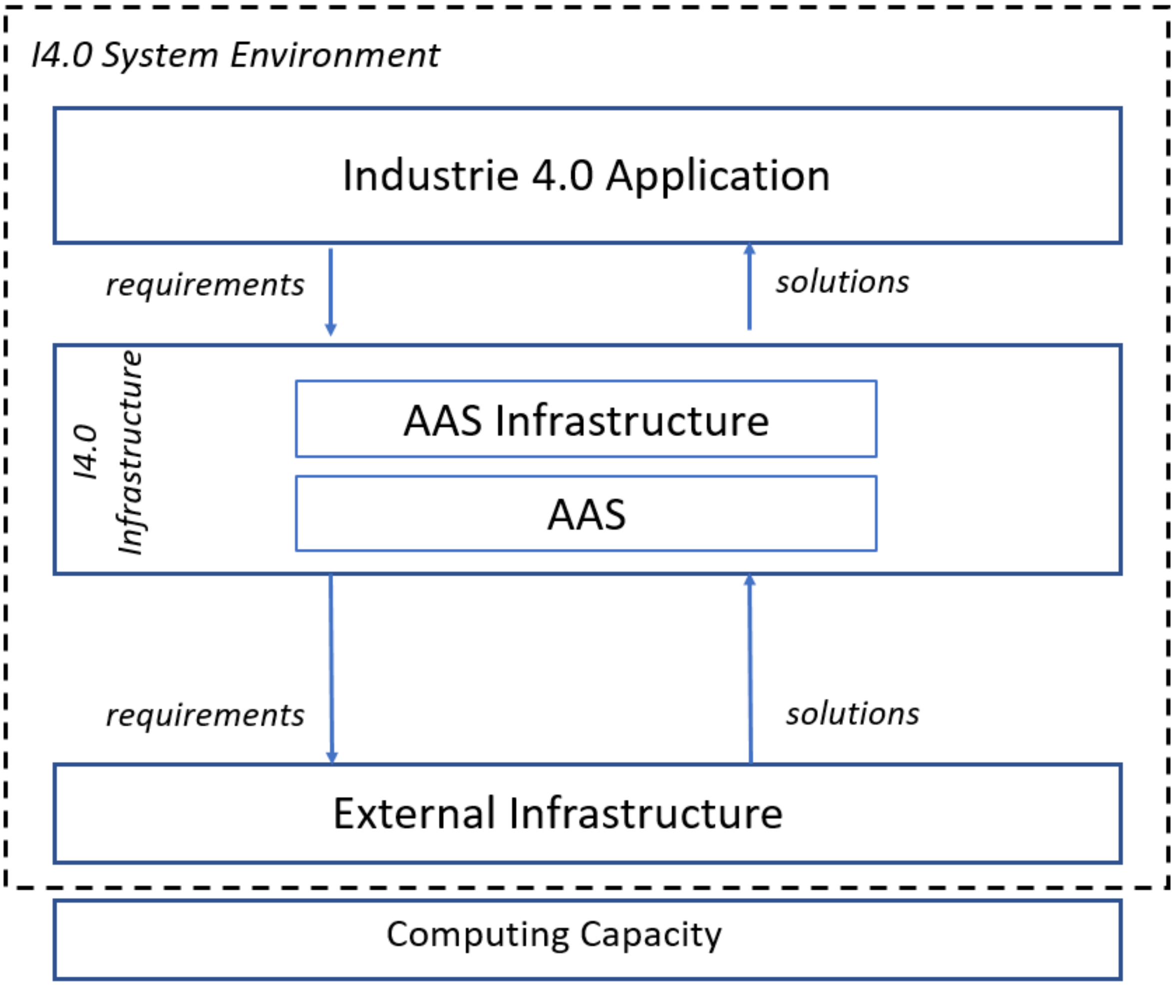


Figure 11 Industrie 4.0 System Environment

9.2.2 Application to I4.0 Service and System Management

For the topic of I4.0 Service and System Management, implementations and instances of asset-related services and infrastructure services may also be seen as assets themselves. This enables to encompass the management of service implementations and service instances as part of an I4.0 System leading to the situation that all interaction patterns and interaction policies that are valid of AAS of physical assets may also be applied to services. No additional concepts are required. This idea is illustrated in Figure 13.

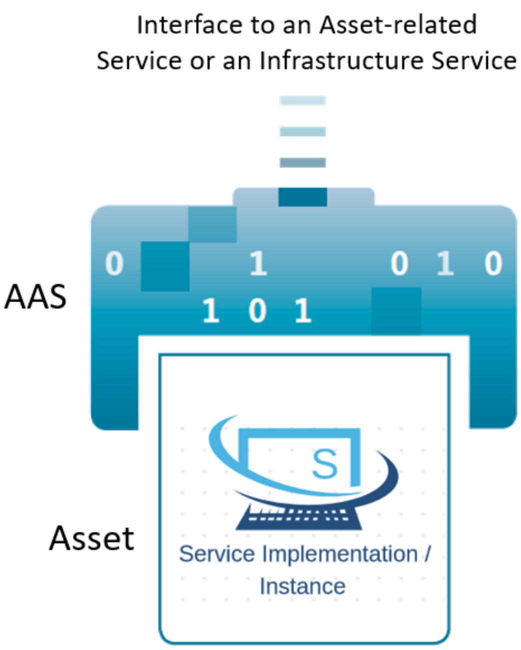


Figure 13 Service Implementations and Instances modelled as assets

Furthermore, in case we consider an I4.0 System as a particular type of asset itself, the AAS-based modelling approach also allows to specify a management interface for "Industrie 4.0 Systems" as a whole. Hence, no additional concept is required to manage I4.0 Systems. It is just required to define AAS sub-models, respectively, and offer their functionality through AAS-compliant management interfaces. These may be considered as a special kind of infrastructure services, too. This idea is illustrated in Figure 12.

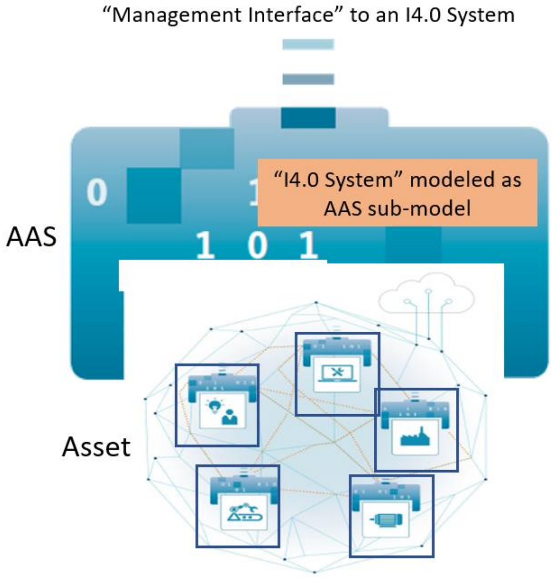


Figure 12 AAS-compliant Management Interface to an I4.0 System

When defining infrastructure services that are systemic relevant for Industrie 4.0, it is essential to define rules and constraints how I4.0 components that offer and use these services may interact.

For these reasons, two more concepts are applied from the DIN SPEC 16593-1 [3] in order to have a terminological framework to express these rules: (interaction) policies and (interaction) domains.

- Interaction Policy
  The concept of an interaction policy allows a system engineer to define rules and constraints. For this purpose, dedicated policy languages exist. An example of such a policy language is XACML, the eXtensible Access Control Markup Language defined by OASIS, for security aspects. XACML may be used for access control in order to express who may access which information element under which conditions.

In general, an interaction policy is defined as "textual or formal specification of constraints or conditions on the interactions between components". To be applied in I4.0 systems policy definitions need to be constrained to a scope of validity. This scope of validity is defined via the concept of an interaction domain.

- Interaction domain
  The concept of an interaction domain allows a system engineer to define the scope of validity of rules and constraints. It is defined as a "set of interacting components" whereby the components refer to I4.0 components in an Industrie 4.0 system. Such an interaction domain may either be defined directly by a list of its constituting components (e.g., list of AAS identifiers), or indirectly by other criteria (e.g., logical, geographical or time-based filters).

Hence, by combining these two concepts, an interaction policy is finally defined as "textual or formal specification of constraints or conditions on the interactions between components that are valid in a specified interaction domain".

Note that in an Industrie 4.0 system several interaction policies dedicated to different aspects (e.g., access control, data usage control, activity logging, negotiation, performance management) may exist in parallel. Consequently, several interaction domains will exist in parallel, possibly but not necessarily overlapping. Interaction domains will typically but conceptually need not necessarily coincide with organizational boundaries, e.g., of an organization or an industrial plant. It may also cover the geographical area of a supply chain or a subset of it.

9.3 Interaction Patterns with and between Asset Administration Shells

There is a general understanding, e.g., described in [5] that, from a stakeholder perspective, different use cases need to be supported that result in different interaction patterns between AAS instances. These are defined on the technology-neutral level and mapped to the technology-specific level (Figure 14).

The following high-level interaction patterns between AAS instances are distinguished:

(1) Type 1 AAS Interactions (Passive): Interactions which need a standardized structuring of information according to the meta-model of an AAS. The exchange of AAS content is defined by a technology mapping of the AAS meta-model (e.g., AASX, XML, JSON or AutomationML). AAS are exchanged between I4.0 system participants in form of a file.

This means, that the I4.0 system participants are decoupled from each other and do not directly interact.

(2) Type 2 AAS Interactions (Re-active): Interactions which need a standardized interface to AAS content to another I4.0 system participant following different interaction patterns and architectural styles [3] such as REST or also request/reply and publish/subscribe as realized by, e.g., OPC UA. For example, a software application can interact with AAS instances if it supports the interfaces of the AAS and if it knows the endpoint of the AAS, e.g., by querying the AAS registry with the AAS ID.

Note: In [8] these interactions are classified as vertical interactions as the involved system participants behave asymmetrically and do play different roles (e.g., client/server, publisher/subscriber) according to the architectural style and its interaction patterns.

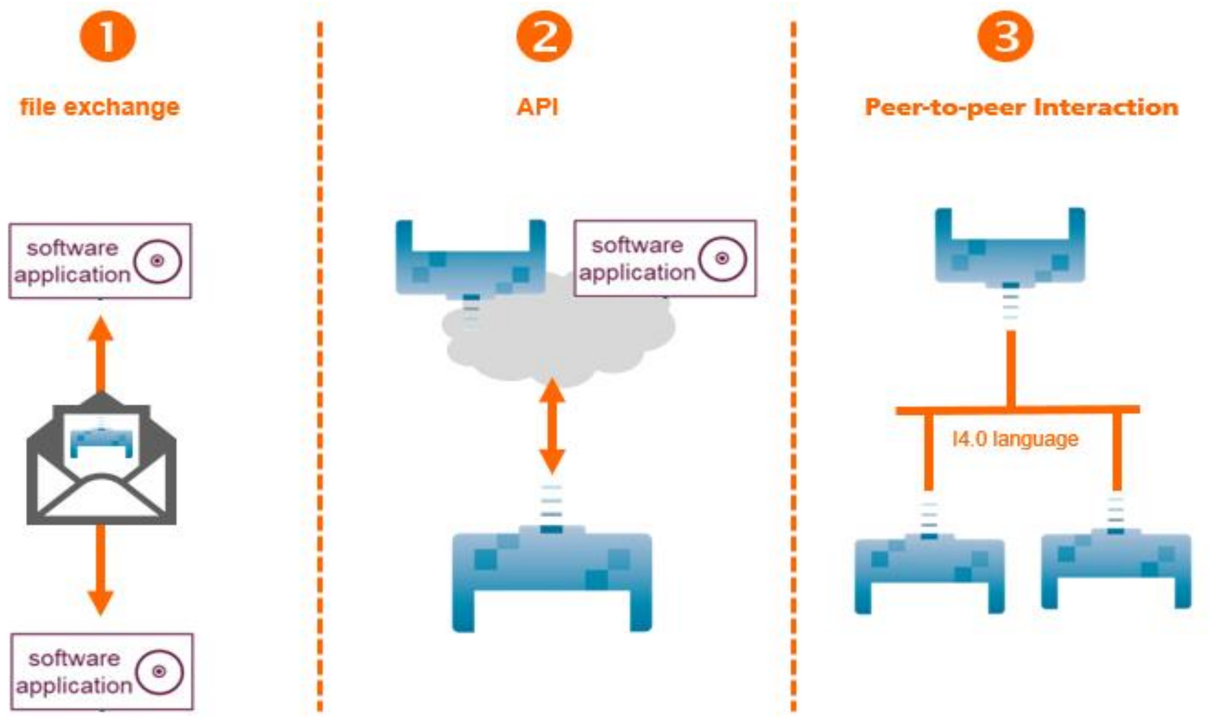


Figure 14 Illustration of typical interaction patterns with and between Asset Administration Shells

(3) Type 3 AAS Interactions (Pro-active): Interactions which need peer-to-peer interactions between AAS of I4.0 components. The ex-change of messages may be structured according to a grammar of an Industrie 4.0 language [9] or may be implemented by using standardized APIs. Type 3 AAS interactions enable I4.0 com-ponents to become pro-active components which can support system integration and system interaction in a peer-to-peer environment between I4.0 components e.g., to enable AAS assisted plug & play scenarios between I4.0 components.

10. Infrastructure Services

10.1 Introduction

The answer to the question „What functionality shall be provided to applications by infrastructure services? “depends on the answers to the related question: “What is the application and what are the common requirements of this application for infrastructure services?

Section 0 states that Industrie 4.0 systems are built of interacting Industrie I4.0 components while I4.0 components consist of an AAS instance, submodels listed in this AAS instance and one associated asset.

In addition to this the following main categories of requirements can be derived from the value proposition as discussed in section 7.

- Unification of information structuring and access to asset related information
- Foundation for an asset-oriented view
- Provision of mechanisms for trusted restriction of usage of information

Consequently, application components interact with I4.0 components by using their AAS interface as the single-entry point to the submodels and asset related services.

As a pre-condition to enable such interactions by using these single-entry points and to support the value proposition several categories of necessary Meta Information are discussed in 10.2.

10.2 Meta-information Categories

Industrie 4.0 components will offer different asset related services in different implementation technologies. It is obvious that not every asset related service of one I4.0 component is able to interact with every asset related service of another I4.0 component. In practice this is even not necessary because interactions take place with a certain goal. This problem is solved today by existing standards and by an engineering using such standards.

It is not likely that all industrial verticals will become harmonized by a small set of standards for e.g., data models, communication protocols or capability profiles. Arguments amongst others are different needs which are served by different standards as well as the installed base already implemented compliant to different standards.

If the concept of the AAS shall fulfil its value proposition than it must provide functional mechanisms to support the simplification of today’s procedures e.g., in engineering. Otherwise, it would just substitute one data modelling technology by another.

The following problem statements are worthwhile to consider when designing infrastructure services:

- Is an application component technically capable to interact with a given instance of a software service?
- Is it required that an application component shall interact with a given instance of a software service?
- Is it allowed that an application component interacts with a given instance of a software service? In particular:
    - Has an application component the right to access to the data of a software service?
    - Under which constraints has an application component the right to use the data provided by a software service?

These problem statements are applicable to many tasks in Industrie 4.0 e.g., in automation but not limited to. Amongst others they formulate core problems on a conceptual level to be solved by classical engineering tasks as well as at runtime.

The Categories of meta information necessary to tackle the problem statements above and to meet the requirements as discussed in clause 7 are explained in detail by using the example of an asset related service.

Hypothesis 1:

Different categories of meta information are necessary to fulfil the value proposition of the AAS.

Figure 15 uses an asset related service as an example of an I4.0 application component to illustrate the concept.

Assuming Hypothesis 1 is valid the same set of meta information is necessary also for AAS as well as for submodels as they are I4.0 application components too.

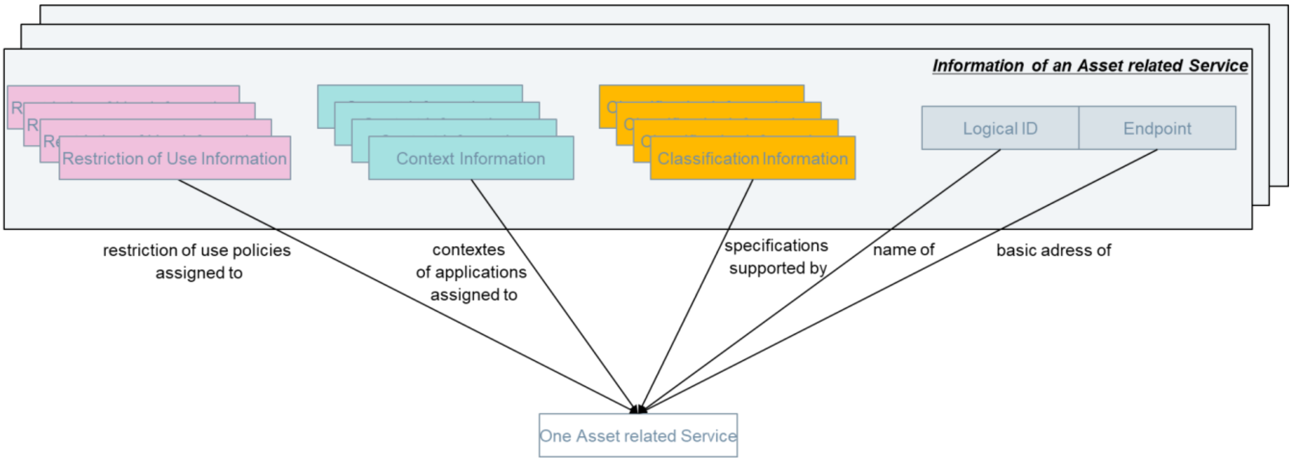


Figure 15 Meta Information which are necessary to fulfil the value propositon

Hypothesis 2:

These different categories of meta information fulfil different purposes and address different parts of the value proposition.

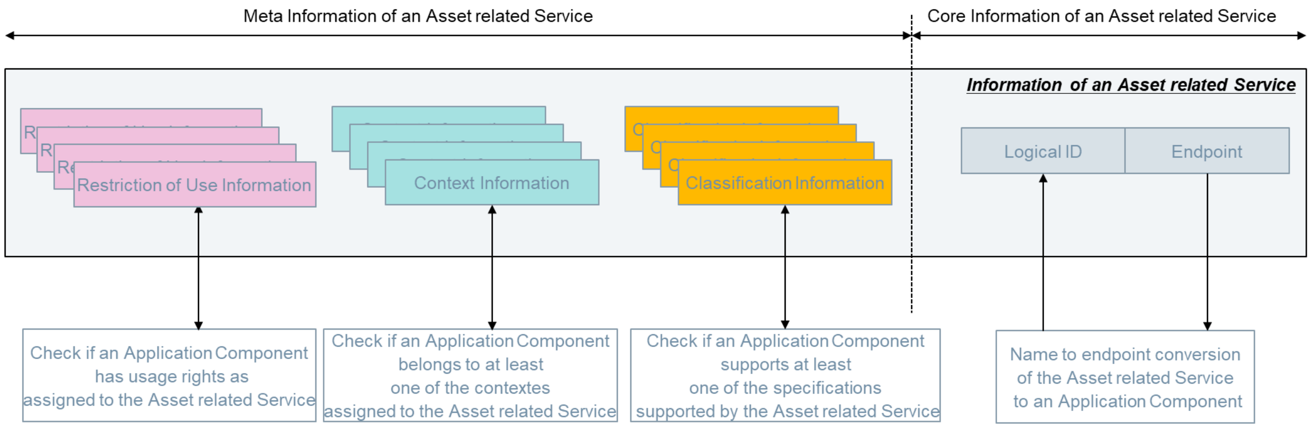
Figure 16 uses an asset related service as an example of an I4.0 application component to illustrate the concept.

Assuming Hypothesis 2 is valid the same set of meta information is necessary also for AAS as well as for submodels as they are I4.0 application components too.

In a summary these meta information determine if an application component:

1. Is technically able to interact with an AAS or an asset related service (Classification Information),
2. Shall interact with an AAS or an asset related service (Context Information),
3. Has usage rights to interact with an AAS or an asset related service (Restriction of use Information) in particular
   a. Is authorized to interact with an AAS or an asset related service (Access Control Information) and if so
   b. Is authorized to use the information provided by an AAS or an asset related service (Usage Control Information).

These conditions to be considered for the interaction of Application Components with submodels, Asset related services as well as with AAS.



Note:
„The Application Component" can be any kind of application supporting the AAS-Interface defined by the interfaces of the set of Infrastructure Services e.g.
- *Any IT application which does not have an own AAS*
- *Another AAS*
- *An Asset related Service referenced in an AAS*

Figure 16 Different purposes of Meta Information to support I4.0 applications and I4.0 systems

Conclusion:

Different categories of infrastructure services are necessary to support different purposes for accessing an AAS, a submodel or an asset related service. They enable the implementation of different aspects of the value proposition of the AAS. This will be explained in more detail for each infrastructure service category in clause 10.8.

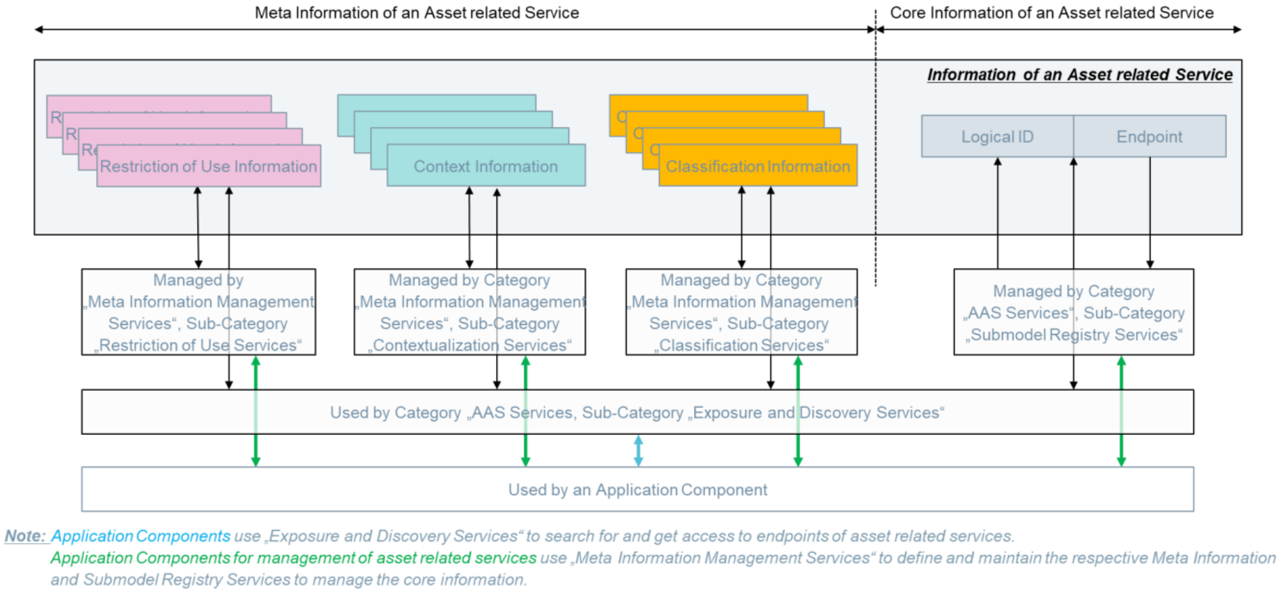


Figure 17 Different kinds of application components use different categories of I4.0 infrastructure services

10.3 Infrastructure Service Categories

The following set of infrastructure service categories defines main areas of functions of the I4.0 computing infrastructure, e.g., to establish an AAS instance, to manage submodels and asset related services and to enable the usage of an Asset Administration Shell.

Figure 18 provides an overview of the infrastructure service categories.

As described in section 0 the identification of the services themselves is out of scope of this document.

In the following sub-sections examples of reusable interfaces are provided according to the Industrie 4.0 Service Model specified in section 0.

All categories of infrastructure services will become motivated as well as described in more detail in the following sections.

It is of special importance to consider that asset related services of Industrie 4.0 components are managed by infra structure services. They are listed in asset administration shells and represented by submodels.

All submodels follow the same harmonized Meta-Model as described in [6].

Meta information as well as core information of asset related services need to be represented by such submodels.

10.4 Conceptual service model of infrastructure services

Figure 19 gives an overview how the infrastructure service categories developed in clause 10.3 and illustrated in Figure 18 are mapped to the service model and to the system model. It illustrates the endpoints and belonging interfaces of the infrastructure service categories.

To fulfill the value proposition of the AAS it is not only necessary to define a standardized set of I4.0 Infrastructure services. Point 2 and 3 of the value proposition require the AAS to be the single-entry point to the services related to an asset. This becomes supported best if infrastructure services are provided by AAS. This principle is also mentioned as “Asset Orientation”.

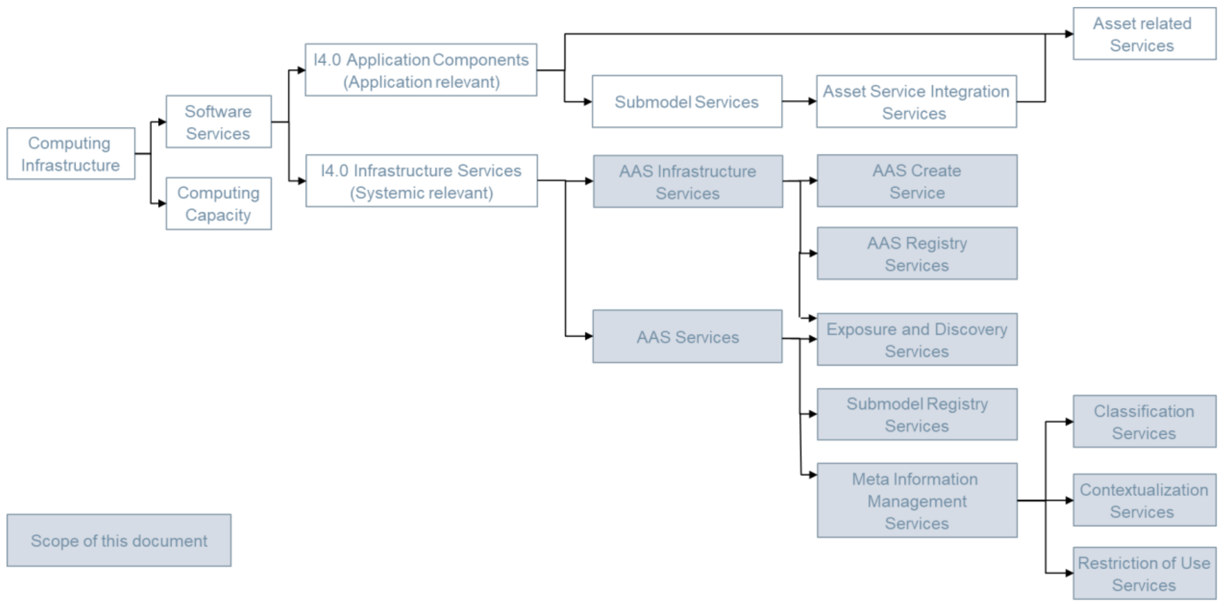


Figure 18 Overview I4.0 Infrastructure Service Categories and I4.0 Application Components

Therefore Figure 19 shows the relationship between I4.0 Infrastructure services in general and a specific subset of it. This subset is named “AAS Infrastructure profile”. The I4.0 Infrastructure services of this profile are characterized by the fact that they assigned to an AAS while the other I4.0 Infrastructure services are not.

These Services are in the scope of this document and this conceptual model is applied to all categories of infrastructure services as described in the following clauses.

It is assumed that Infrastructure services are not assigned to AAS only if it is indispensable for the required functionality e.g., to create an AAS.

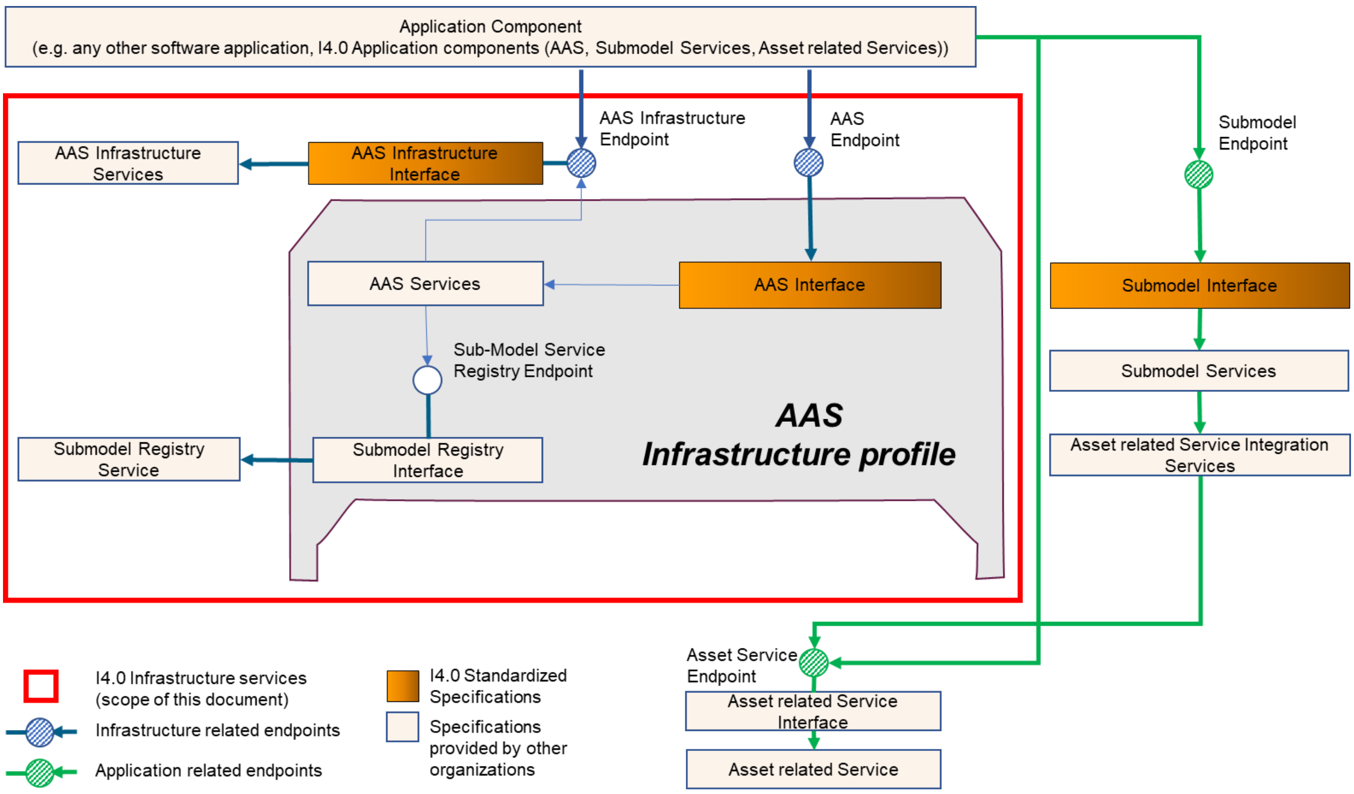


Figure 19 Conceptual Model I4.0 Infrastructure Services

10.5 Conceptual service model of I4.0 infrastructure services and I4.0 Application components

Figure 20 enhances the conceptual model of clause 10.4 by the application profile of the AAS.

It shows how the concept of I4.0 Infrastructure Services and I4.0 application components fit together in an AAS to build I4.0 Systems and I4.0 Applications.

However, the AAS application profile is not part of this document. As explained in clause 8.2 only those software services are considered as infrastructure services which provide a uniform syntax as well as a uniform semantic to application relevant services.

While the syntax of the meta model of the submodel is uniform the semantic is not and differs from application to application. If necessary, the standardization of the semantic of submodels may be done by the communities of the respective stakeholders at their discretion.

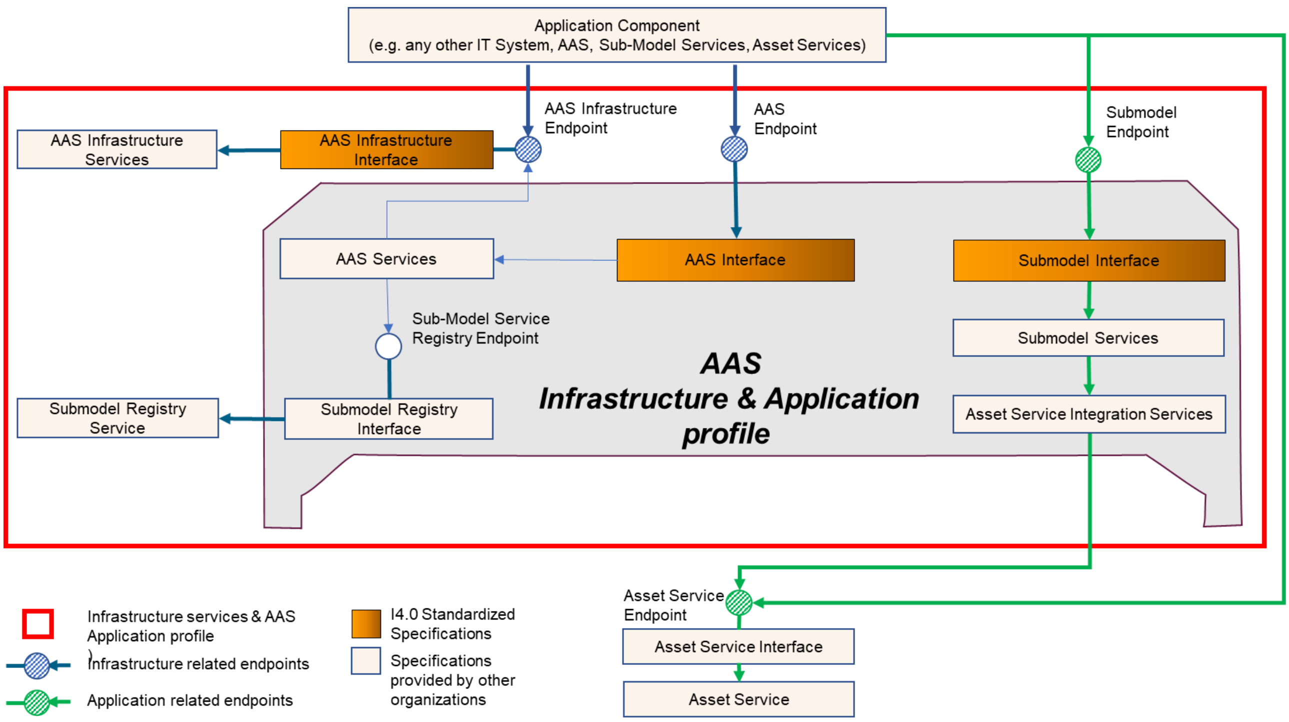


Figure 20 Conceptual Model I4.0 Infrastructure Services and I4.0 Application Components

10.6 AAS Infrastructure Services

10.6.1 Motivation

AAS infrastructure services are necessary to create AAS and to make them findable.

They lay the basis for any usage of an AAS in Industrie 4.0 applications as well as in Industrie 4.0 systems.

They ensure that each AAS is globally biunique identifiable.

AAS infrastructure services are accessible via AAS Infrastructure Interface.

AAS Infrastructure Services are those I4.0 Infrastructure services which are not assigned to an AAS. That's why they don't belong to the AAS Infrastructure profile.

10.6.2 AAS Infrastructure Interface

The AAS Infrastructure Interface is accessible by the "AAS Infrastructure endpoint".

The interface is used to access:

- AAS Create Service
  This service is the AAS_Create service.

  It shall return a globally biunique AAS identifier.

  Once created it can never be deleted nor re-used for identification of other AAS.

  It is the identifier of one AAS instance over its whole lifecycle.

- AAS Registry Services
  Examples of AAS_Registry Services are:
  - Registration of AAS instances in AAS registries
  - De-Registration of AAS instances in AAS registries
- AAS Exposure and discovery Services
  Examples of Exposure and Discovery of AAS instances in AAS registries are:
  - Search for one AAS using its Logical ID. This Service will expose the endpoint of the AAS.
  - Search for AAS using AAS meta information. This Service will expose a list of the endpoints of all AAS consistent with the AAS meta information.

10.6.3 Support of Use Cases

AAS Infrastructure Services lay the foundation of any use case described in [1] Chapter "Activities".

10.6.4 Support of the Value Proposition of the AAS

AAS Infrastructure services primarily support point 1 and 2 of the value proposition. They enable

- Unification of information structuring and access to information,
- Asset orientation in I4.0 systems
- Creation of single-entry points to information which is related to an asset

10.7 AAS Services

10.7.1 Motivation

Controlling the way an application component gets access to a submodel or an asset related service is one of the most important tasks of the ASS.

The purpose of AAS Services is the management of asset related information by a set of infrastructure services which is used in relation to an AAS instance. AAS Services belong to the AAS Infrastructure profile.

10.7.2 AAS Interface

The AAS Interface is accessible via the AAS endpoint.

This interface is used to access:

- Submodel Registry Services for the AAS instance
  - list and delist of submodels to or from an AAS instance
  - registration and de-registration of submodels listed in an AAS instance to or from a Submodel Registry

- Meta-information Management Services for the AAS instance
  - Add, modify, and delete Restriction of Use Information of the AAS instance
  - Add, modify, and delete Context Information of the AAS instance
  - Add, modify, and delete Classification Information of the AAS instance

- Meta-information Management Services for submodel instances listed in one AAS instance.
  - Add, modify, and delete Restriction of Use Information of submodel instance
  - Add, modify, and delete Context Information of submodel instance
  - Add, modify, and delete Classification Information of submodel instance

- Meta-information Management Services for asset related services referenced by submodels listed in one AAS instance.
  - Add, modify, and delete Restriction of Use Information of asset related service
  - Add, modify, and delete Context Information of asset related service
  - Add, modify, and delete Classification Information of asset related service

- Exposure and discovery services for submodels listed in the AAS instance
  - Search for submodels within an AAS using its Logical IDs. This Service will expose the endpoint of the submodel.
  - Search for submodels within an AAS using asset related service meta-information. This Service will expose a list of endpoints of submodels listed within one AAS instance consistent with the submodel meta-information.
  - Searching for all submodels consistent with a defined set of submodel meta-information in all AAS consistent with a defined set of AAS Metainformation. This Service will expose a list of endpoints of all submodels consistent with the submodel meta-information found in all AAS consistent with the AAS meta-information.

- Exposure and discovery services for asset related services referenced by submodels of the AAS instance
  - Search for asset related service within an AAS using its Logical IDs. This Service will expose the endpoint of the asset related service.
  - Search for asset related services within an AAS using asset related service meta-information. This Service will expose a list of endpoints of asset related services referenced by submodels within one AAS instance consistent with the asset related service meta-information.
  - Searching for all asset related services consistent with a defined set of asset related service meta-information in all AAS consistent with a defined set of AAS meta-information. This Service will expose a list of endpoints of all asset related services consistent with the asset related service meta-information found in all AAS consistent with the AAS meta-information.

10.7.3 Support of Use Cases

The AAS Interface supports the use cases described in [1].

Note: The Use cases are directly taken from [1] Chapter “Activities”. They are examples and not comprehensive.

10.7.4 Support of the Value Proposition of the AAS

AAS services primarily support point 1 and 2 of the value proposition. They enable

- Unification of information structuring and access to information,
- Asset orientation in I4.0 systems
- Single entry points to information which is related to an asset

Use Case Name	Description
Design and Integration of Asset Administration Shells	**Activity “Scoping and modeling”** Task 1 “Definition of the asset related service registry (declaration of associated asset related services)” Task 2 “Definition resp. modeling of the asset related services”: Task 3 “Definition of initial setup”: role software engineer
Design and Integration of Asset Administration Shells	**Activity “Implementation, deployment, and test”** Task 3 “Testing the asset related services in conjunction with the overall system consisting of software applications and asset related service registries based on the computing infrastructure” Task 4 “Notification of all software applications and asset related services with an interest in the new asset related service registry and asset related services”

10.8 Meta-information Management Services

10.8.1 Motivation

Meta-information management for AAS submodels and asset related services supports many requirements in Industrie 4.0 as explained in clause 10.2.

There are no dedicated endpoints for interfaces of meta-information management services defined because operations of such interfaces are made available by the AAS interface via the AAS endpoint. AAS Services use the interfaces of meta-information management services. However, it is important to discuss the purpose of meta-information and to ensure that meta-information can be assigned to AAS to submodels as well as to asset related services.

Meta-information Management Services belong to the AAS infrastructure profile and can be divided in three sub-categories.

10.8.2 Classification Service sub-category

Motivation

The purpose of Classification is to indicate that the interface of an Application Component is implemented according to a given specification. Such specification may be e.g., an IEC standard or a company proprietary specification. Insofar Classification determines if an Application Component is in general interoperable with other Application Components. In easy words Classification enables a verification if Application Components fit together technically

Classification can be done in many ways. Examples are:

- A declaration of a supplier that his product is conformant to a given standard.
- A certificate of a notified body that the product is tested and conformant to a given standard.
- A self-description e.g. a semantic annotation of a software service that it is conformant to a given specification.

The first two cases are examples for implicit usage of this classification. Interacting applications are set up in a way that an explicit check if the components fit together is done before the system is set up. Often this check is part of the engineering.

However, in Plug and Play scenarios as explained in detail in clause 13 in this document it is important that an Application Component can search for other Application Components and verify upfront if they could technically interact. An important pre-condition is a fit with the Classification. Therefore, an explicit Classification which can be verified at runtime is necessary. Classifying attributes must be defined and assigned to Application Components.

It is of special importance that this is even valid for the Infrastructure Services for AAS as well as for the application components.

Infrastructure services for AAS will develop over time. This will take place based on evolving standardization of such services. E.g., for backwards compatibility reasons with APIs it is important to classify them.

Classification of asset related services is very useful to support Step 1 (Plug). Many of the current standards don ‘t support explicit classification. Asset related services could use it to search for other classified asset related services.

Infrastructure services should support the Classification of Infrastructure Services and asset related services.

Different aspects of Classification might be useful.

For higher level search, it is often of interest which asset related services according to a given Classification Scheme are available.

For those cases asset related services could be classified e.g., according RAMI4.0.

For lower-level search, it is mainly of interest if asset related services are available which fit functionally. For those cases asset related services could be classified according to the vertical standards they support.

In this sense Classification Services manage Classification Metainformation for AAS registered in AAS-Registries, to submodels listed in AAS and to asset related services referenced by AAS.

Classification Interfaces (examples)

AAS Classification Interface

This Interface is used to add and delete Classification Information to AAS.

It is expected that there will be a development of AAS specifications and the belonging Infrastructure Services. This will lead to different versions of AAS specifications in the field in parallel. To support proper interaction with all of these different versions a classification of AAS instances themselves is necessary.

- Add Classification ID to AAS
- Delete Classification ID from AAS

Submodel Classification Interface

This Interface is used to add and delete Classification Information to submodels.

It is expected that there will be a development of submodel specifications and the belonging services. This will lead to different versions of submodel specifications in the field in parallel. To support proper interaction with all these different versions a classification of the submodel itself is necessary.

- Add Classification ID to submodel
- Delete Classification ID from submodel

Asset related service Classification Interface

This Interface is used to add and delete Classification Information to asset related services.

It is assumed that an asset related services can be implemented according to any given specification and that each of these specifications can be deployed in the field in different versions and in parallel. To support proper interaction with all these different versions of specifications a classification of the asset related service is necessary.

- Add Classification ID to asset related service
- Delete Classification ID from asset related service

Support of Use Cases

The Classification Interfaces support the use cases described in [1].

Use Case Name	Description
Design and Integration of Asset Administration Shells	**Activity "Scoping and modeling"** Task 2 "Definition resp. modeling of the asset related services" Task 3 "Definition of initial setup"
Design and Integration of Asset Administration Shells	**Activity "Implementation, deployment, and test"** Task 3 "Testing the asset related services in conjunction with the overall system consisting of software applications and asset related service registries based on the computing infrastructure":
Usage of Assets	**Activity "Acquisition and commissioning of an asset: physical world asset"** Task 1 "Acquisition of the asset" Task 4 "Integration of the asset administration shell"

Note: The Use cases directly taken from [1] Chapter "Activities". They are examples and not comprehensive.

Support of the Value Proposition of the AAS

Classification Services mainly support point 2 of the value proposition.

- They support the ease of system integration and reduce the engineering effort
- They support an AAS assisted plug & play and enhance the flexibility in I4.0 systems as described in clause 13.
- They support the integration of the installed base

10.8.3 Contextualization Services sub-category

Motivation

In general, contextualization is "the process of identifying the data relevant to an entity based on the entity's contextual information" whereby contextual information "characterizes the situation of an entity in interaction with other entities".

Transferred to an Industrie 4.0 system or an Industrie 4.0 application, the purpose of contextualization is to determine that and how application components belong together under defined circumstances, e.g., to express, enforce or check that they are following the same interaction policies.

The circumstances itself define the frame of the context. The application components belonging to a context and the way how they interact and perform their functions define the content of the context. In easy words Contextualization enables a validation if application components belong together to build a common function. It is of special importance that this is valid for the Infrastructure Services as well as for the application components.

In this sense Contextualization enables system synthesis and system interaction from an application point of view. It may use explicit classification of application components as a pre-condition.

There are many examples of context types, e.g.

- hierarchical contexts,
- functional contexts
- organizational contexts.

Figure 21 shows an example of a hierarchical context of an asset.

The asset with its belonging parts defines the frame of the context. The asset related services of the AAS instances and the way how they interact following the rules of the Hierarchy of the Asset and define the content.

Figure 21 also explains the two concepts "context" and "relation" which belong together as follows:

- Context defines that application components belong together.
- Relations define the content of the context. In other words, relations describe not that but how application components belong together.

That means that application components have a defined set of relations within a given context at a time.

A change of application component and / or relations in a given context changes the content of the context or may generate a new context.

Semantics e.g., the vocabulary of relations are valid within the context the relations belong to. They may not be valid or expressed differently in other contexts. E.g. The relation „belongs_to" may have the same meaning in context 1 as relation „is_part_of" has in context 2. In this sense each context defines its own semantic.

Application Components may be part of different contexts in parallel and therefore may have different sets of relations in parallel. Imagine a car which has an air nozzle in a door. In its „hierarchy context" the air nozzle „belongs to" the door. In its „functional context" it „is part of" the Air Conditioning System.

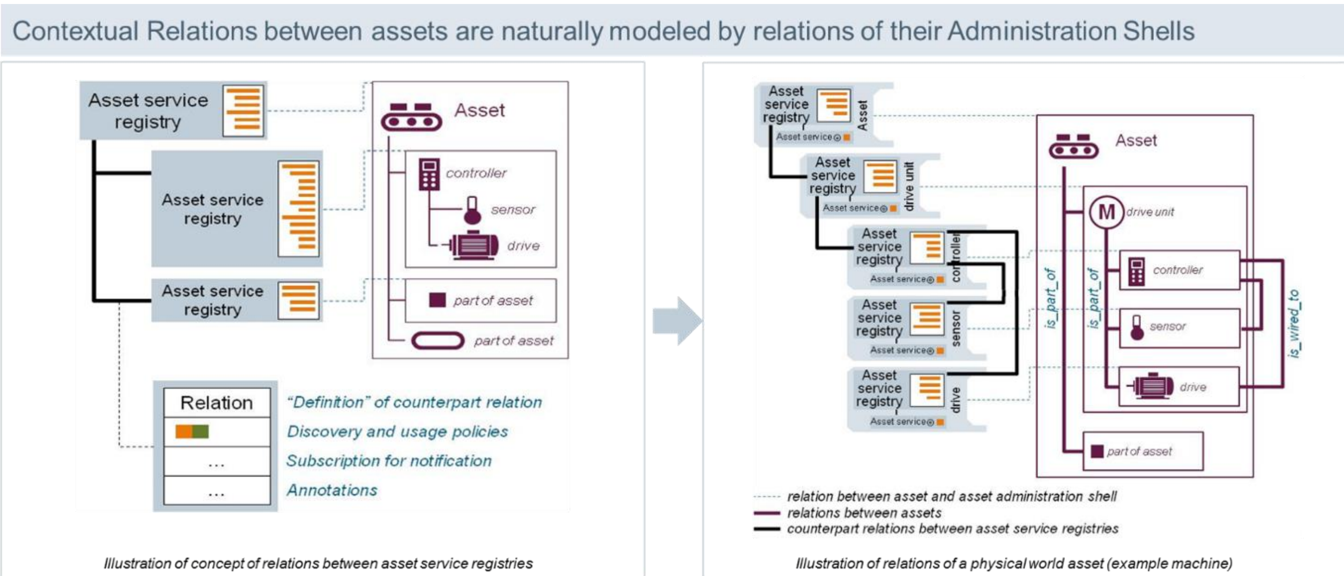


Figure 21 Contextual Relations

As of today, Context Information is very often embedded in engineering systems. Some use it internally and implicitly; others provide standardized explicit context information.

However, in Plug and Play scenarios as explained in detail in Annex 4 in this document it is important that an application component can search for other application components and verify upfront if they shall functionally interact. The most important pre-condition is a fit with the Contextualization. Therefore, an explicit Contextualization which can be verified at runtime is necessary. Contextualization attributes therefore must be defined and assigned to application components e.g., to AAS, to submodels and to asset related services.

Contextualization attributes can be expressed in many ways and it is a design decision for an application designer to select.

Examples for Contextualization attributes are:

- Tags of plane text of a given vocabulary.

  This is a simple case where application components know upfront based on their implemented algorithms which relations with other application components they have. They only need to search for such application components tagged with a matching text to determine the application components belonging to the same context

- Context expressions using JSON-LD
- Context expressions as defined by ECLASS
- Link to an ontology using a given vocabulary using OWL.

  OWL is an advanced case where application components can reason on their own role in a context, reason on other application components they have relations with and which ones and then search and find these application components.

Let's take the car-example from above to illustrate the differences between semantic tagging and semantic reasoning and to support the understanding of contextualization in general.

By using a plain text to express a context the Air Conditioning controller would search for application components of the Air Conditioning System using a text tag like "Air Conditioning-System". It will find them but needs to know upfront e.g., by programming, that it will find air nozzles and how to interact with them. In a first step this gives a lot of flexibility at runtime because the Air Conditioning System must not be engineered to the today's necessary detail and mechanisms like late bindings can be used in the field.

This means that application components only need to know upfront what needs to be done and will find the other application components at runtime.

By using an Ontology to express a context the Air Conditioning controller only needs to know that it is an Air Conditioning controller. It can reason on the ontology to find the other application components in our case air nozzles which belong to the Air Conditioning System of the given exemplar of the car and can also reason on how to interact with them.

This means that application components can reason at runtime what are the other application components and what needs to be done with them.

Automation ML can be seen as another technology providing explicit and standardized context in engineering processes. It is one example how context information is expressed today.

Figure 22 shows an example for organizational contexts. Every level of decomposition of it constitutes a context the underlaying levels belong to.

Contextualization Services manage Context Metadata to AAS registered in AAS-Registries to submodels listed in AAS and to asset related services referenced by submodels.

Contextualization Interfaces (examples)

AAS Contextualization Interface

This Interface is used to add and delete Context Information to AAS

- Add Context ID to AAS
- Delete Context ID from AAS

Submodel Contextualization Interface

This Interface is used to add and delete Context Information to submodels

- Add Context ID to submodel
- Delete Context ID from submodel

Asset related service Contextualization Interface

This Interface is used to add and delete Context Information to asset related services

- Add Context ID to asset related service
- Delete Context ID from asset related service

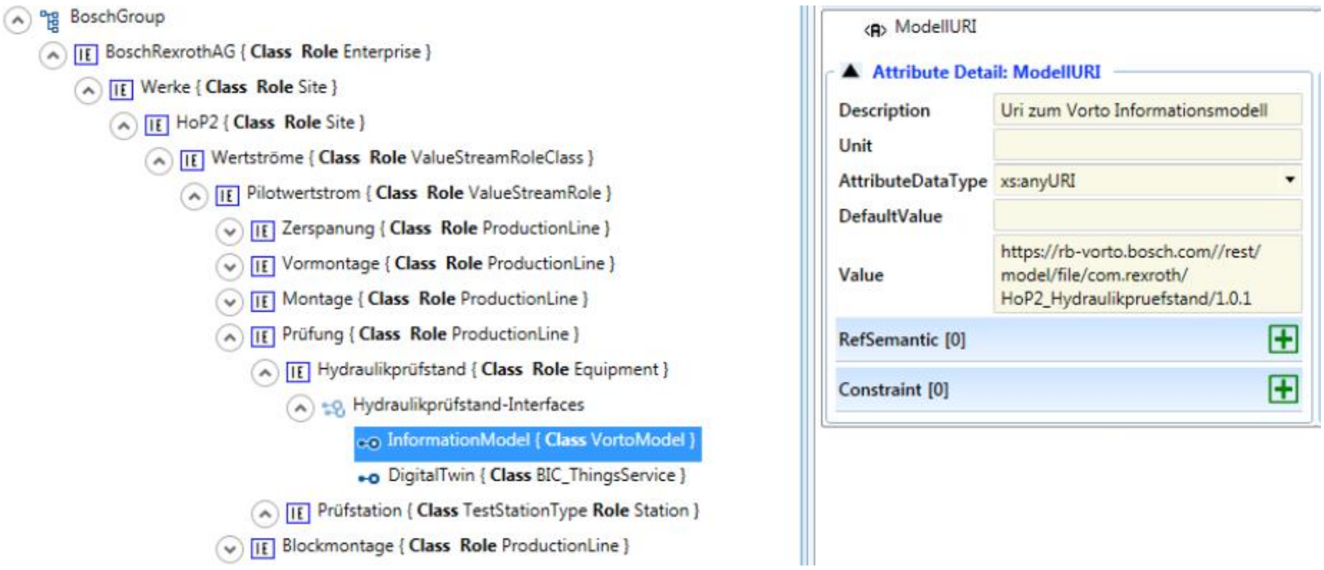


Figure 22 "Hydraulikprüfstand" belongs to Context "Prüfung"

Support of Use Cases

The Contextualization Interfaces support the use cases described in [1].

Use Case Name	Description
Design and Integration of Asset Administration Shells	**Activity “Scoping and modeling”** Task 1 “Definition of the asset related service registry (declaration of associated asset related services and relevant relations to other assets)” Task 3 “Definition of initial setup”
Design and Integration of Asset Administration Shells	**Activity “Implementation, deployment, and test”** Task 3 “Testing the asset related services in conjunction with the overall system consisting of software applications and asset related service registries based on the computing infrastructure”: Task 4 “Notification of all software applications and asset related services with an interest in the new asset related service registry and asset related services”
Usage of Assets	**Activity “Acquisition and commissioning of an asset: physical world asset”** Task 3 “Integration and commissioning of the asset (integration and commissioning in the real world)” Task 4 “Integration of the asset administration shell”
Standardization of Asset related services and Relations	**Activity “Standardization of asset related services and relations”** Task 1 “Definition of asset related services mandatory resp. optional for an asset (including implementation guidance for the asset related services, commissioning guidance for assets and application policies for software applications) to be compliant with the standard” Task 2 “Definition of relations mandatory resp. optional for an asset (including design guidance for service registries and application policies for software applications) to be compliant with the standard“
Miscellaneous Applications	**Activity “Handling of intelligent assets with integrated asset administration shell”** Task 1 “Initial commissioning (identical to ‘plug&produce’ case in activity ‘Acquisition and commissioning of an asset (physical world asset)’)” Task 2 “Removal from the computing infrastructure (see activity ‘Modification of an asset (physical world asset)’, whereby task 3 must ensure that all asset related services, which should be furthermore available, are now deployed on an available computing resource of the computing infrastructure)”: Task 3 “Re-integration into the computing infrastructure (see activity ‘Modification of an asset (physical world asset)’, whereby task 3 must ensure that the asset related services are now back-deployed on the computing resource provided by the asset)”

Note: The Use cases directly taken from [1] Chapter “Activities”. They are examples and not comprehensive.

Support of the Value Proposition of the AAS

Contextualization Services mainly support point 2 of the value proposition.

- They support the ease of system integration and reduce the engineering effort
- They support an AAS assisted plug & play and enhance the flexibility in I4.0 systems as described in clause 13
- They support the integration of the installed base

10.8.4 Restriction of Use Services sub-category

Motivation

To understand the underlying principles of restriction of use it is crucial to clarify two foundational concepts. These two concepts are “ownership” and “possession”. They are defined as below.

- Ownership: “Most comprehensive control of an object allowed by a legal system”.
- Possession: “Actual power of disposal of an object”.

Restriction of Use is a process where an owner of an object defines usage rights for objects he owns and grants such usage rights at his discretion to possessors he selects.

The rights to define and grant usage rights are only limited by the spawn of control allowed by the legal system.

The result of the process is always a relationship between owner and possessor w.r.t. an object and expressed in usage rights of the possessor granted to him by the owner.

There are many different areas where restriction of use is required. Some examples are:

- Law related Restriction of Use e.g., usage rights to comply with export control and IP Protection Regulations
- Business related Restriction of Use e.g., usage rights to enforce terms and conditions of Contracts, Licenses and Payments
- Organization related Restriction of Use e.g., usage rights to enable a Function of a role or a context of a role
- Security related Restriction of Use e.g., usage rights to ensure IT-Security, Information Integrity, and Privacy

Restriction of Use can be divided in two main aspects which are explained in more detail in the sub-clauses below. These aspects are:

a) Access Control
is the process to authorize an application component to interact with another application component

b) Usage Control
is the process to authorize an application component to use the information provided by another application component in defined ways and for defined purposes.

Note that the Restriction of Use services need an (external) infrastructure that is being specified and provided within an Industrie 4.0 System Environment but outside of the I4.0 Infrastructure, see Figure 11. An example of such an external infrastructure will be provided by GAIA-X, possibly built-in concepts and technologies provided by the International Data Spaces (IDS).

Access Control

The purpose of access control is to protect system resources such as assets 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.

In this document, we focus on access control to resources in an Industrie 4.0 System, i.e., the data and operation elements of AAS instances as part of an I4.0 Component. Other facets of access control, e.g., with respect to laws and business-related resources (such as export control, IP Protection, contracts, payments…) are not considered per se. They are only relevant in the sense that the access control policies deployed in an Industrie 4.0 system should be typically derived from a security management process as determined in the ISO/IEC 27000[1] series or IEC62443 series for industrial automation and control systems.

Access Control relies upon Identity Management and can only be successfully implemented in a secure environment to limit the access and subsequently usage of information. Corresponding rules which are determined and defined by the owner of the information or by organizations which have permissions to do so.

Trusted identities are the foundation of three step approach which general applies to all areas of access control

1. Identification: process of an entity stating its own identifier to other entities.
2. Authentication: process of an entity to validate the identifier stated by another entity.
3. Authorization: process to grant access rights to an entity by using the result of the authentication process

Together, these processes constitute the Identity and Access Management (IAM) as a base concept for the design of trusted and secure applications. Main purpose of IAM is to ensure that access to resources can be restricted to those roles which have the respective access rights in a way that the requirements of the information owner are met.

Usage Control

Usage control is one cornerstone of data sovereignty as the capability of a natural person or legal entity for exclusive self-determination regarding their data goods. Usage control is an extension to traditional access control. It is about the specification and enforcement of restrictions regulating what must (not) happen to data. Thus, usage control is concerned with requirements that pertain to data processing (obligations), rather than data access (provisions). Usage control is relevant in the context of intellectual property protection, compliance with regulations, and digital rights management [11].

Transferred to the AAS concept in Industrie 4.0, usage control refers to the question how control can be maintained and enforced about the usage of data (of an AAS instance) by application components or other AAS instances after access has been granted to AAS data and operations.

Policy Management Interface (examples)

This Interface is used to define and manage an access and usage control domains and policies and assign them to policy domains. It encompasses the following operations:

- Create/Update Domain (list of AAS-IDs): domainID
- Read/Delete Domain (domainID): list of AAS-IDs
- Create/Update Policy (policyDescription): policyID
- Read/Delete Policy (policyID): policyDescription
- Assign PolicytoDomain (policyID, domain-ID)
- Decouple PolicytoDomain (policyID, domain-ID)

[1] In ISO/IEC 27000 access control is defined as "means to ensure that access to assets is authorized and restricted based on business and security requirements".

- SearchPolicy (policyCriteria)
- RetrievePolicyDescription

This operation is used to validate policies according to the protection goals for an I4.0 System or a domain within an I4.0 System.

- ValidatePolicy (Policy-ID, Domain-ID, I4.0 System

Support of Use Cases

The Policy Management Interface supports the use cases described in [1].

Note: The Use cases directly taken from [1] Chapter "Activities". They are examples and not comprehensive.

Use Case Name	Description
Design and Integration of Asset Administration Shells	**Activity "Scoping and modeling"** Task 4 "Definition of access rights managed by the asset related service directory (access rights of software applications to asset related services, access rights to relations of this asset to other assets)" Task 5 "Definition of access and usage policies of the asset related services": **Activity "Implementation, deployment, and test"** Task 2 "Deployment of the asset related services including following all requested policies, e.g., with respect to IT security" Task 3 "Testing the asset related services in conjunction with the overall system consisting of software applications and asset related service registries based on the computing infrastructure"
Design and Integration of a Software Application	**Activity "design, implementation, deployment, and test"** Task 2 "Implementation of the software application including a 'defensive' programming due to the possibility that access rights for asset related services may change during operation" Task 3 "Deployment of the software application including following all requested policies, e.g., with respect to IT security" Task 4 "Test of the software application": role software engineer
Usage of Assets	**Activity "Acquisition and commissioning of an asset: physical world asset"** Task 3 "Integration and commissioning of the asset (integration and commissioning in the real world)": role owner of the asset Task 3.1 The asset provides own computing resources: in this case the computing resources of the asset have to be integrated into the computing infrastructure including following all requested policies, e.g., with respect to IT security, using the capabilities provided by the computing infrastructure. **Activity "Modification of an asset: physical world asset"** Task 1 "Modification and commissioning of the asset (modification and commissioning in the real world)" Task 1.1 The modification affects own computing resources of the asset: The modification with respect to the integration into the computing infrastructure has to be executed by following all requested policies, e.g., with respect to IT security. Task 2 "Redesign of the asset administration shell of the asset according to the intended modification" Task 3 "Integration of the redesigned asset administration shell, especially suitably deploying all requested asset related services"

Support of the Value Proposition of the AAS

Access and Usage Control Services partially supports point 2 and mainly support point 3 of the value proposition.

- They support the protection of investment
- They enable system integrity and system protection
- They enable IP protection and commercialization of assets and asset related services
- They support an AAS assisted plug & play and enhance the flexibility in I4.0 systems as described in annex 13

10.9 Exposure and Discovery Services

10.9.1 Motivation

The purpose of Exposure and Discovery to enable Application Components to search and find endpoints of other Application Components using a defined set of criteria.

In I4.0 such endpoints are basic addresses of Interfaces of:

- AAS
- Submodels
- Asset related services

The criteria for exposure and discovery are defined by the meta-information assigned to AAS, submodels and asset related services by meta-information Management Services.

In scenarios where endpoints are not explicitly made available to Applications Components (e.g., by Engineering) Exposure and Discovery Services provide the necessary search mechanisms

For this purpose, exposure and discovery service make use of meta-information assigned to AAS, submodels and asset related services. As described in clause 10.8.

Examples:

- Searching for AAS
    - An Application Component looking for specific administration shells can use the AAS Infrastructure interface or the AAS Interface depending on the search task to search for an AAS matching defined search criteria. The Interface returns the logical ID of a set of administration shells matching the described search – if the application has appropriate access rights to see existing AAS.
    - With these logical IDs, the Application Component can request AAS-Endpoints which will be provided if the Application Component has appropriate access rights to interact with the AAS.

- Searching for Submodels
    - An Application Component looking for specific submodels can use the AAS interface to search for submodels matching defined search criteria. The Interface returns the logical ID of a set of submodels matching the described search – if the Application Component has appropriate access rights to see existing submodels.
    - With these logical IDs, the Application Component can request submodel-Endpoints which will be provided if the software Application Component has appropriate access rights to interact with the submodels.

The criteria to search and find endpoints are expressed as set of Metadata as described in clause 10.2.

These meta-information are managed by meta-information Management Services which add them to the respective information sets in registries.

Exposure and Discovery Services use registries loaded and managed by AAS infrastructure Services or AAS services.

Exposure and Discovery Services act as the final gate keeper and decide if a logical ID or an endpoint is delivered to the searching Application Component. They do so by using the set of Metadata assigned to the endpoint and a corresponding set of Metadata delivered by the searching Application Component.

E.g.

- Exposure and Discovery Services enable application components to expose and search for properties and values of AAS and find identifiers of the meta-model entities
- Exposure and Discovery Services enable application components to find AAS and submodels which belong to the same context
- Exposure and Discovery Services enable application components to find AAS and submodels they have usage rights on
- Exposure and Discovery Services enable application components to find AAS and submodels which support one or more given specifications

10.9.2 Exposure and Discovery Interfaces (examples)

There are no dedicated endpoints for Exposure and Discovery Services defined because operations of such interfaces are made available by the AAS Infrastructure interface and by the AAS interface.

They are part of the AAS Infrastructure profile if used via the AAS Interface.

They are not part of the AAS Infrastructure profile if used via the AAS Infrastructure Interface.

AAS Exposure and Discovery Interface

This Interface is used to expose and discover AAS

- Search for one AAS using its Logical ID. This Service will expose the endpoint of the AAS.
- Search for AAS using AAS meta-information. This Service will expose a list of the endpoints of all AAS consistent with the AAS meta-information.

Submodel Exposure and Discovery Interface

This Interface is used to expose and search submodels listed in an AAS

- Search for submodels within an AAS using its Logical IDs. This Service will expose the endpoint of the submodel.
- Search for submodels within an AAS using submodel meta-information. This Service will expose a list of endpoints of submodels within one AAS consistent with the submodel meta-information.
- Searching for all submodels consistent with a defined Set of submodel meta-information in all AAS consistent with a defined Set of AAS meta-information. This Service will expose a list of endpoints of all submodels consistent with the submodel meta-information found in all AAS consistent with the AAS meta-information.

Asset related service Exposure and Discovery Interface

This Interface is used to expose, and search asset related services referenced by submodels listed in an AAS

- Search for asset related services within an AAS using its Logical IDs. This Service will expose the endpoint of the asset related services.
- Search for asset related services within an AAS using asset related service meta-information. This Service will expose a list of endpoints of asset related services

within one AAS consistent with the asset related service meta-information.

- Searching for all asset related services consistent with a defined Set of asset related service meta-information in all AAS consistent with a defined Set of AAS meta-information. This Service will expose a list of endpoints of all asset related services consistent with the asset related service meta-information found in all AAS consistent with the AAS meta-information.

10.9.3 Support of Use Cases

The Exposure and Discovery Interfaces support the use cases described in [1].

Use Case Name	Description
Usage of Assets	**Activity "Acquisition and commissioning of an asset: physical world asset"** Task 4 "Integration of the asset administration shell"
Miscellaneous Applications	**Activity "Handling of intelligent assets with integrated asset administration shell"** Task 1 "Initial commissioning (identical to 'plug&produce' case in activity 'Acquisition and commissioning of an asset (physical world asset)') Task 2 "Removal from the computing infrastructure (see activity 'Modification of an asset (physical world asset)', whereby task 3 must ensure that all asset related services, which should be furthermore available, are now deployed on an available computing resource of the computing infrastructure) Task 3 "Re-integration into the computing infrastructure (see activity 'Modification of an asset (physical world asset)', whereby task 3 must ensure that the asset related services are now back-deployed on the computing resource provided by the asset)
Plug and Play	**Search for Application Components to interact with in a common application** Task 1 Search for Application components with a given characteristics (Context, Rights of Use, Classification) Task 2 Evaluate Context and Classification Information Task 3 Enforce Rights of Use Policies Task 4 Deliver Logical ID or Endpoints

Note: The Use cases "Usage of Assets" and "Miscellaneous Applications" are directly taken from [1] Chapter "Activities". They are examples and not comprehensive.

10.9.4 Support of the Value Proposition of the AAS

Exposure and Discovery Services support all three points of the value proposition.

- They implement a unifying concept to structure and access asset related services.
- They enable the ease of system integration and reduce the engineering effort
- They enable an AAS assisted plug & play and enhance the flexibility in I4.0 systems
- They enforce a reliable restriction of use of information of an asset and implement the single-entry point to it

11. Annex 1 (informative): Meta-data Concepts

Some of the Infrastructure service categories discussed in this document fall into the categories of services in distributed systems that deal with meta-data. As related terms and concepts are not used consistently all over the computer science literature and product portfolios, it is essential to give precise definitions in the scope of Industrie 4.0.

First, we define the following terms and concepts in the context of an AAS and its computing infrastructure (non-exhaustive list):

- Registry
- Directory
- Catalogue
- Repository
- Dictionary

The following definitions are proposed in the context of an AAS computing infrastructure:

AAS Directory

collection of AAS identifiers and/or AAS submodel identifiers and associated subset of AAS meta-data content to be gathered for searching for AAS. The meta-data content is dependent on the purpose of the AAS Directory.

Notes:

- An AAS Directory does not need to be official.
- An AAS Directory does not need to be comprehensive, i.e., it may contain only a subset of the AAS that are deployed in a given domain, and not all meta-data content elements need to be necessarily filled.
- An AAS Directory provides an open API to search for and manage AAS entries. The open API is described in an Industrie 4.0 specification that falls into the AAS Implementation View.
- An AAS Directory may be operated by any type of organization and company.
- AAS Directories may be organized in networks or hierarchies.

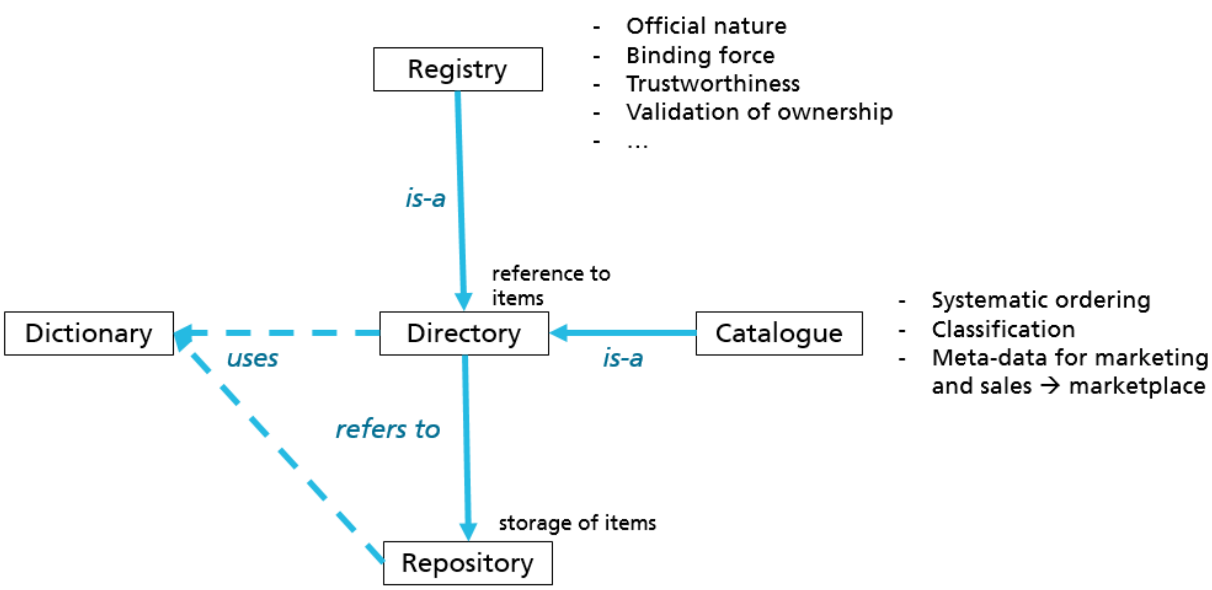


Figure 23 Relationship between Basic Concepts and Terms for Meta-data Services

AAS Registry

AAS Directory operated by designated stakeholders that are legitimated to associate the directory entries with some statements of liability and trustworthiness, e.g., about identity, jurisdiction, ownership or security.

Notes:

- An AAS Registry may have to undergo a process of certification to be specified by the Platform Industrie 4.0.
- An AAS Registry may be "official" if it follows a governance regime of a public administration or government.

AAS Catalogue

An AAS Catalogue is an AAS Directory operated by some organization or company that provides additional context to the directory entries such as classification, systematic ordering or additional meta-data for marketing and sales.

Notes:

- An AAS catalogue may be used to support marketplace scenarios for AAS, e.g., order-based production[2].
- The metadata elements of AAS catalogues are out of scope of Industrie 4.0 standardization, just their core as a variant of an AAS Directory is relevant. Hence, this concept is just mentioned for the sake of completeness.

Dictionary

Oxford Dictionary[3]: A book or electronic resource that lists the words of a language (typically in alphabetical order) and gives their meaning, or gives the equivalent words in a different language, often also providing information about pronunciation, origin, and usage.

Notes:

- In an AAS computing infrastructure the "words" of a language are to be interpreted as the concepts defined in a domain-specific or generic dictionary such as eCl@ss or IEC CDD (Common Data Dictionary).

[2] Examples are the IIC testbed Smart Factory Web (https://www.smartfactoryweb.de) or the IOTA Industry marketplace (https://industrymarketplace.net/).

[3] https://en.oxforddictionaries.com

12. Annex 2 (informative): Deployment Scenarios

Plug & Produce requires that application components, that are potentially provided by third parties, can be introduced to any compliant system with a minimum of integration efforts.

This requires the specification of required interactions and their compliant implementation from various perspectives:

Structural/technical interoperability (connectivity): Being able to transfer data payload from one to another application component

Syntactical interoperability: Being able to identify distinct (semantically assessable) information elements and data structures in the transferred payload and extract them for further processing.

Semantic interoperability: Being able to interpret the extracted information elements semantically correct.

Organisational interoperability: Being able to organize interacting processes in an effective and efficient way.

In the lifecycle of a third-party application component for seamless integration into system-interactions that contribute to its system-goals, several interactions are required:

- E.g., Identify requirements and develop application component
- E.g., Transfer application component to system owner
- E.g., Integrate application component into system
- Application component to use asset related service
  - Application component lookup relevant AAS
    - Organisation of system components
  - Application component find submodel service
    - Identify semantics
  - Application component find asset related service
    - Identify syntax and structure
  - Application component call asset related service
    - Usage of functionality
- E.g., Application component to be removed from system

From the viewpoint of the application component, it is essential to identify the interactions with the system and its components (infrastructure, AAS, submodels, asset related services) – to be able to define a desired degree of interoperability.

Furthermore, to minimize the constraints for the implementation of application or system components, the technology, design, or deployment of the components should be left to the left to the providers of the components, as long as they keep to the Interaction and Interface specification for the specific system components.

To give an idea for different scenarios to implement the structures defined through AAS in Detail series and Usage/Functional View documents we depict 4 Deployment scenarios for clustering AAS concepts into services that might be deployed to different nodes running the service. These scenarios are not exhaustive and support dedicated qualities and – if implemented according to the Interface specifications – allow to be mixed.

The application component itself has no need to know in which scenario it is integrated. There is no dependency of the application component to the deployment scenario and the only information that is depending on the specific deployment scenario is managed through the management of endpoint-addresses.

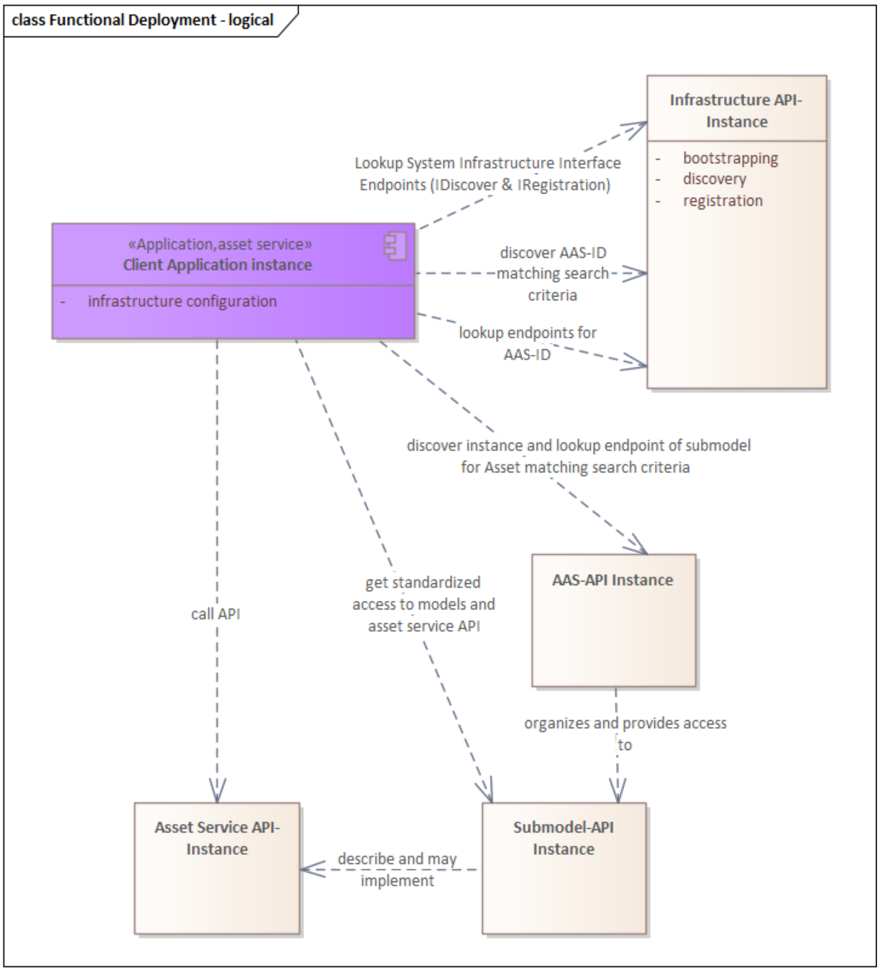


Figure 24 Application component and System interactions

Scenario 1: “Monolithic” deployment

This first scenario assumes that most of the Implementations required to build a System of Asset administration shells is implemented through one generic AAS System provider service. This Service offers all Interfaces of the logical components that are implemented within that Service.

One exception to the inclusion of functionalities in the AAS System provider service are external asset related services that might offer their functionality to the system through proprietary protocols (like e.g., machinery or other legacy installations).

Pros:

- Very efficient for systems that have a single provider for all system components

Cons:

- High degree of technical dependencies between all AAS-system elements

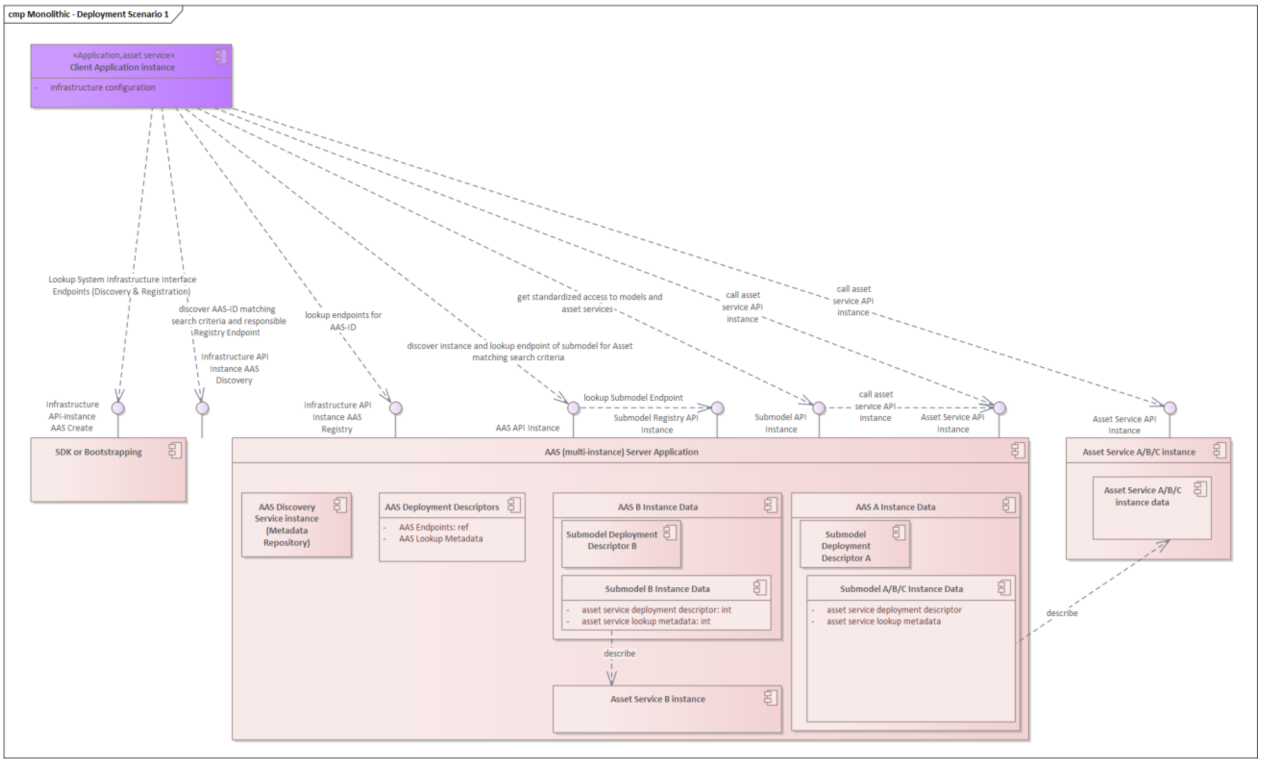


Figure 25 monolithic deployment scenario

Scenario 2: "Self contained" deployment

In this Scenario it is expected that all AAS-dependant concepts might be implemented within a dedicated system component comprising the implementatio of AAS and Submodel-concepts. This allows a provisioning of AAS as "self contained systems" delivering all AAS-dependant implementations as one component ensuring consistency throughout the dependant AAS-artefacts.

Pros:

- Flexible implementation of Asset Administration shells and integration of Implementations of different Providers deployed to different Systems possible. High consistency of directly AAS dependent implementations.

Cons:

- Higher Complexity in managing dependencies of different service providers and dependencies to compliance to interface implementation.

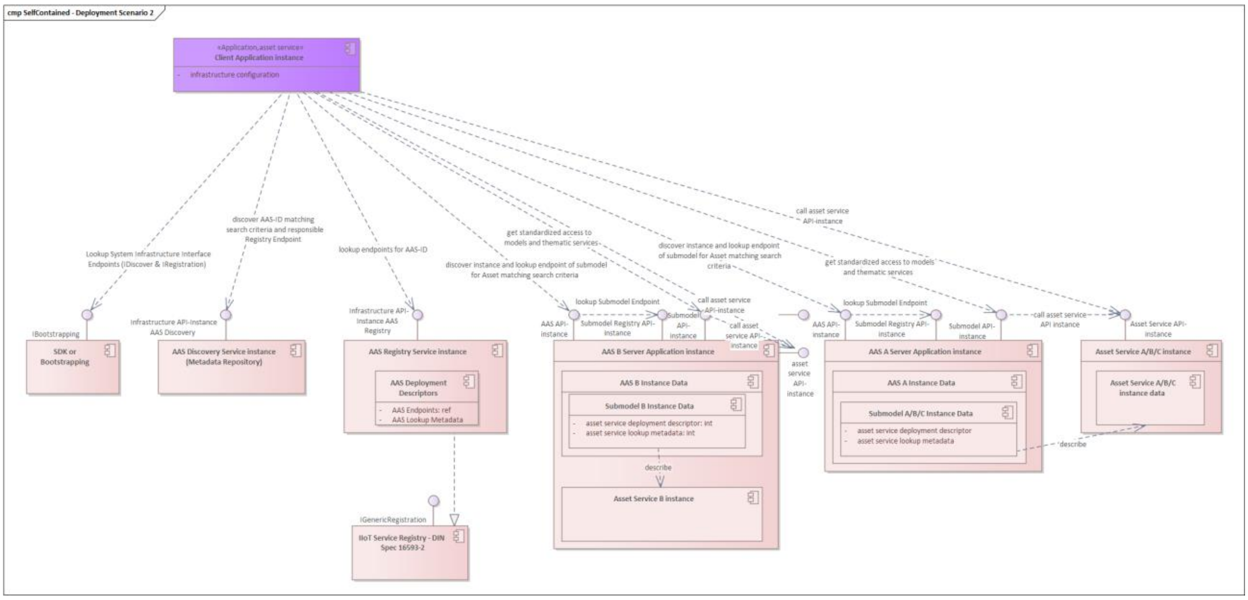


Figure 26 self contained deployment scenario

Scenario 3: “Distributed” deployment

In This deployment scenario all addressable components are separately implemented and can be distributed on different network nodes. This allows a very flexible compilation of elements provided by various implementation partners – still integrating all these components through the integration capabilities of the AAS-concepts.

Pros:

- Maximum flexibility in selection of component services and deployment to different Service runtime environments (e.g., on-device)

Cons:

- Higher Complexity in managing dependencies of different service providers and dependencies to compliance to interface implementation.

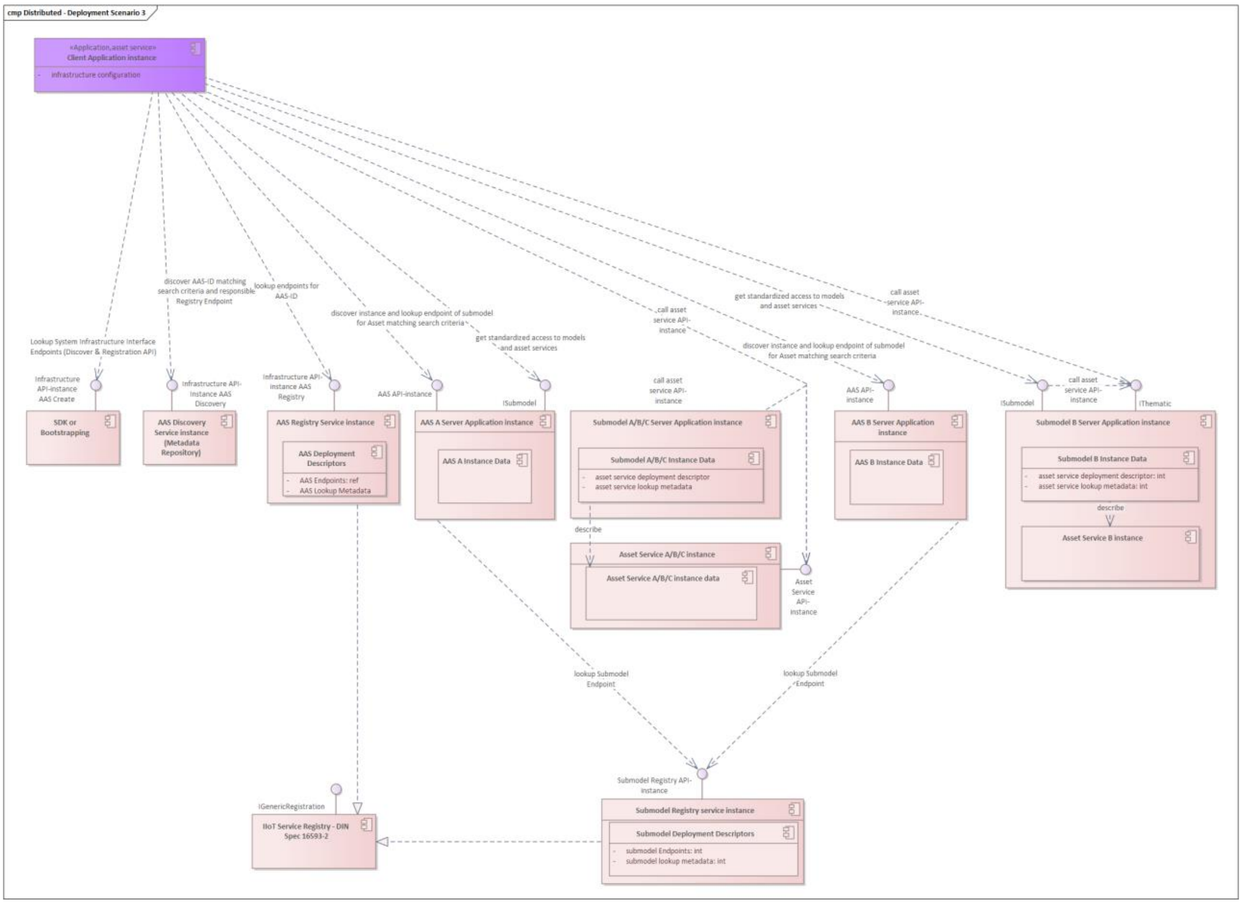


Figure 27 distributed deployment scenario

Scenario 4: “Nested” deployment

This scenario is very similar to Scenario 3 but assumes, that the management of all AAS in a System is managed by a “higher level” Asset Administration Shell which comprises references to subordinate AAS.

Pros:

- Maximum flexibility in selection of component services and deployment to different Service runtime environments (e.g., on-device)
- Whole system behaves and can be accessed through standard AAS-Interactions.

Cons:

- Higher Complexity in managing dependencies of different service providers and dependencies to compliance to interface implementation.

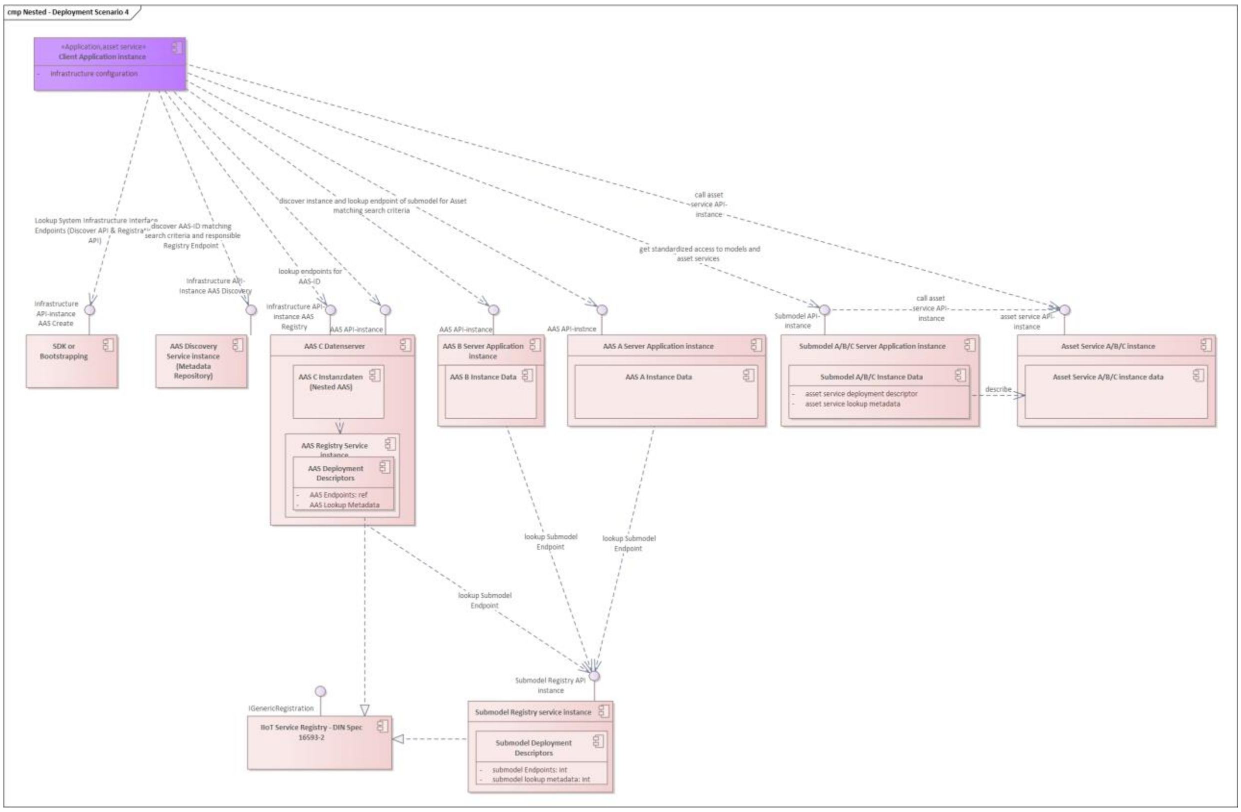


Figure 28 nested deployment scenario

13. Annex 3 (informative) Use Case "Plug and Produce"

13.1. Motivation

Plug and Produce is one of the most promising visions of Industrie 4.0. It is mentioned in various publications and driven by many different goals and stakeholders. Different aspects of interoperability are necessary to achieve Plug and Produce. Examples are:

- Making the product part of the Reference Model RAMI4.0
- Harmonization of Information Models in different Standardization bodies e.g., IEC Digital Factory
- Usage of Industrial Communication standards e.g., SDN Standards
- Adding Semantics e.g., Ontologies using semantic Web technologies
- Definition of capability profiles e.g., OPC UA companion specification

All of them are important to achieve Plug & Produce functionality. The objective of Plug and Produce is to lower the engineering effort and to improve flexibility, adaptability and self-organization of production processes as well of products during the entire lifecycle.

13.2 Use of Infrastructure Services

The main question is how the AAS and the belonging set of Infrastructure Services can support Plug and Produce. This evaluation is important because it is part of the value proposition of the AAS to ease system integration and to reduce engineering effort in even heterogeneous systems.

> Note: At this point we replace "Plug and Produce" by Plug and Play" to give it a broader scope

Plug and Play can be seen as an application pattern using the AAS and the Infrastructure Services as a basis.

Following the principle of the three Types of AAS, we propose to conceptual divide Plug & Play into two distinctive steps and to use Re-active and Pro-Active AAS accordingly.

(1) **Step 1: „Plug"**
Needs to be standardized on Industrie 4.0 level and defines the usage of Infrastructure Services of the computing infrastructure to implement AAS of Type 2 and in particular Type 3.

(2) **Step 2: „Play"**
Is standardized in respective verticals on application level and is not part of the Infrastructure Services of the computing infrastructure to implement AAS. However in order to enable „Play" in Type 3 scenarios those standards may need to be enhanced and adapted to some Infrastructure Services offered by AAS Implementations.

The qualitative step provided by Type 3 AAS consists of a standardized support of

Step 1: „Plug"

independent from the standards used in

Step 2: „Play".

I4.0 Infrastructure services enable AAS assisted *Plug & Play* (important part of the value proposition of I40)

- ***Step 1***: Explore, search and find asset related services, which need to interact in a defined context in order to fulfil a common function. (*Plug*)
- ***Step 2***: Execute the defined interaction of asset related services in order to achieve the common function in the defined context. (*Play*)

Therefore I4.0 Infrastructure services need to deliver a standardized functionality necessary to implement ***Step 1*** (I4.0-conformant).

The Interaction of ***Step 2*** is done by asset related services according their implementation standards e.g. OPC/UA and is independent from the AAS (Functional Profiles, Application-conformant).

Figure 29 Illustration of the Two Step Approach for Plug & Play

This approach gives a maximum of flexibility to design, deploy and integrate Industrie 4.0 applications as well as Industrie 4.0 systems by taking advantage from the concept of Industrie 4.0 Component.

Additionally, it provides a mechanism to make assets of the installed base Industrie 4.0 components and let them become part of those systems to a degree which depends on the Type of AAS available for those assets.

Furthermore, the management of AAS and its asset related services shall enable the integration of heterogeneous vertical technologies for asset related services over their lifetime and integration of the installed base.

It is obvious that asset related services which e.g. belong to different life cycle phases of an asset will use different technologies and standards for their implementation such as:

- JT File Download via ftp for CAD Data
- OPC/UA for reading process values
- Profinet Profiles for functionality at run time
- Etc.

All these different examples of asset related service may be referenced at the same AAS at the same time.

Independent and in addition the AAS must enable assets of the installed base to become Industrie 4.0 components.

Figure 30 on the next page illustrates the concept of an AAS assisted Plug and Play scenario.

The black arrows describe Step 1 (Plug).

The colored arrows describe different and in parallel existing possible Steps 2 (Play).

The colors shall express that the asset related services with the same color are implemented in the same technology and belong to the same context from an application point of view.

Plug and Play of the red marked asset related services in Figure 30 can be described as an example as below:

In AAS assisted Plug and Play scenarios the red asset related service in AAS 2 will use the exposure and discovery services provided by AAS services of AAS 2 to let it search for the right asset related service to interact with. Therefore, it will deliver the meta-information assigned to it for the search procedure to AAS2.

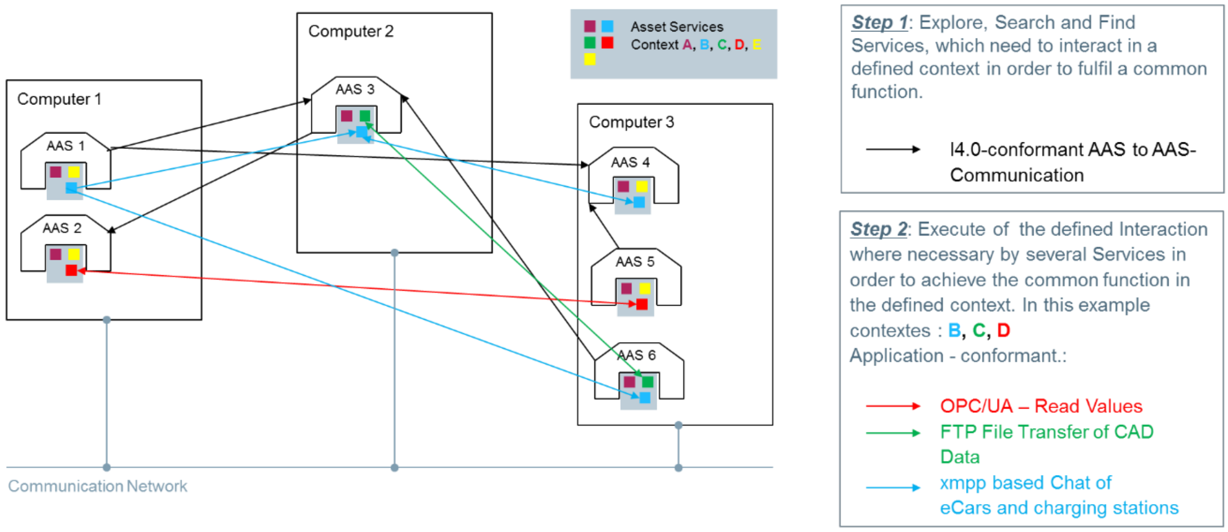


Figure 30 Illustration of Plug and Play

The search request of the red asset related service of AAS 2 will use the meta-information such as Context Information, Classification Information and Restriction of use Information assigned to the asset related service.

The Exposure and Discovery Services of AAS 2 will search by asking the other AAS for it and eventually find AAS 5 where the corresponding red asset related service is located in.

Both AAS will exchange information according to the Restriction of use Information applied to.

Amongst others AAS 5 will check:

- If AAS2 has access rights to AAS 5.
- If the meta-information of the searching red asset related service of AAS 2 match the meta-information of the red asset related service of AAS5 and then AAS2 and AAS5 will finally exchange the endpoints of their asset related services.

AAS 2 will pass the endpoint to its requesting asset related service and AAS 5 may pass the corresponding endpoint of the searching asset related service to the asset related service which was found by request.

Step 1 (Plug) is performed after this exchange of endpoints. From now on the red services know each other and can perform Step 2 (Play) without any further assistance of the AAS.

In this way AAS and Infrastructure Services support Step 1 (Plug) by a minimum of meta-information required and ensure that only those asset related services interact which have permission.

Editorial Team

Torben Miny, Lehrstuhl für Prozessleittechnik, RWTH Aachen University

Guido Stephan, Siemens AG

Dr. Thomas Usländer, Fraunhofer IOSB

Jens Vialkowitsch, Robert Bosch GmbH

This publication was produced by the working group on „Reference Architectures, Standards and Norms“ (Plattform Industrie 4.0)

This publication was produced by the working group on „Reference Architectures, Standards and Norms“ (Plattform Industrie 4.0)