PLATTFORM INDUSTRIE 4.0

DISCUSSION PAPER

Usage View of Asset Administration Shell

In collaboration with

STANDARDIZATION COUNCIL INDUSTRIE 4.0

ロボット革命イニシアティブ協議会
Robot Revolution & Industrial IoT Initiative

VDI VDE VDI/VDE Society Measurement and Automatic Control

Imprint

Publisher
Federal Ministry for Economic
Affairs and Energy (BMWi)
Public Relations
11019 Berlin
www.bmwi.de

Editorial responsibility
Plattform Industrie 4.0
Bertolt-Brecht-Platz 3
10117 Berlin

Design
PRpetuum GmbH, Munich

Current as at
February 2019

Printed by
BMWi

Image credits
Getty Images – Yuichiro Chino (title)
Getty Images – Westend61 (p. 4)
Getty Images – D3Damon (p. 7)
Getty Images – Monty Rakusen (p. 13)
Getty Images – Enis Aksoy (p. 29)
Getty Images – Monty Rakusen (p. 32)

You can obtain this and other brochures from:
Federal Ministry for Economic Affairs
and Energy (BMWi)
Public Relations
Email: publikationen@bundesregierung.de
www.bmwi.de

Central ordering service:
Tel.: +49 30 182722721
Fax: +49 30 18102722721

Content

Foreword

In the context of Industrie 4.0 the concept of asset administration shell is widely discussed. Nevertheless, a comprehensive view from an application point of view is still missing. However, this is necessary so that a broad community better understands the objectives of this concept to be able to generate benefits from its usage.

We like to thank the members of the working group "modeling examples" of VDI/VDE Society Measurement and Automatic Control (VDI/VDE-GMA) Technical Committee 7.21 "Industrie 4.0 – Terms, Reference Models and Architecture Concepts" and the "Use Case Task Force" in the International Standardization Action Group, Robot Revolution & Industrial IoT Initiative (RRI), for having taken this approach and various members of working groups of the Plattform Industrie 4.0 for their contribution and open discussion. These activities are part of the Germany-Japan cooperation.

The presented elaboration is an important step for completing a common view to a core concept of Industrie 4.0 and to derive requirements for necessary standardization activities. Its results are supported by the strong commitment of VDI/VDE GMA, the Plattform Industrie 4.0, and the Robot Revolution & Industrial IoT Initiative thanks to the open minded and integrating procedure chosen.

We also would like to thank the Standardization Council Industrie 4.0 (SCI4.0) for initiating and orchestrating the activities and partners within the project GoGlobal Industrie 4.0. This project strives for global harmonization and interlinkage of German Industrie 4.0-concepts with regional partnerships and strategic standardization development organizations.

Prof. Dr. Ulrich Epple, Johannes Kalhoff

Prof. Emeritus Dr. Eng. Fumihiko Kimura, Toru Ishikuma

Introduction

Background

In the context of Industrie 4.0 for some time now, the recognition is aware that – in addition to a technology-driven bottom-up approach – one must complement the topic of digitization in manufacturing industries also by a "use case"-driven top-down approach. In such a top-down approach the starting point are possible future business scenarios, from which more technical use case descriptions are derived. These use cases are the basis to derive new products, solutions, and services as well as standardization requirements.

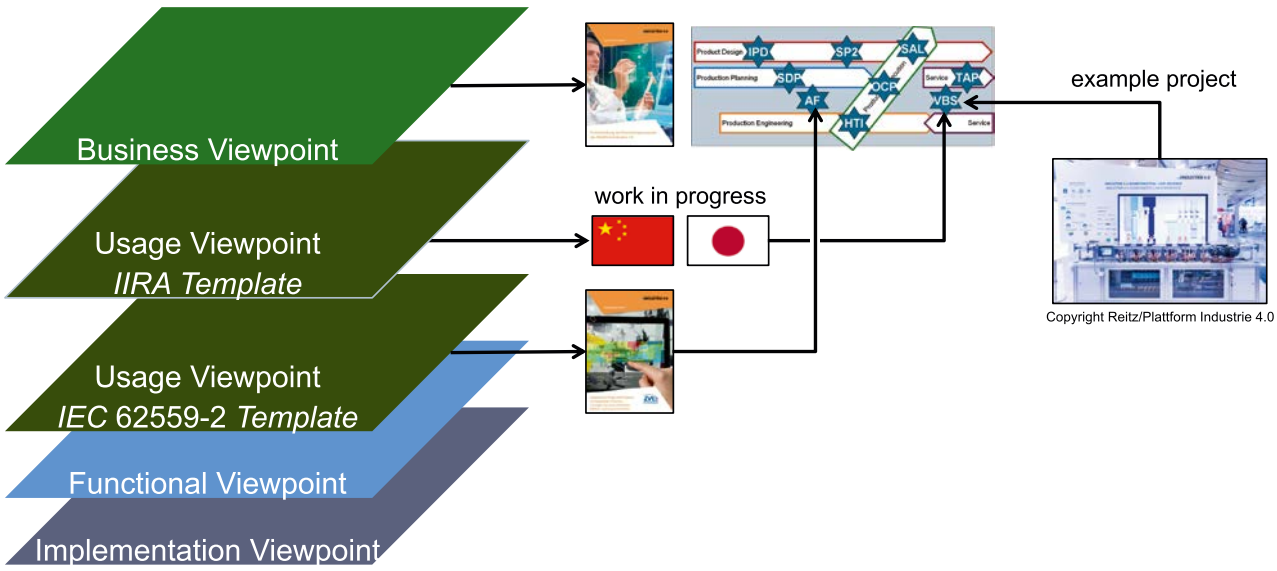
This approach has been applied several times in the past. In particular, the Industrie 4.0 application scenario Value-based Service was considered and described from different perspectives. This application scenario introduces a business perspective into the technical discussion about the Internet of Things resp. Industrial Internet resp. Connected Industries, see [1]. Specifically, a business view was described in [2], a usage view in [3], and a more detailed usage view with references to functional domains in [4].

The knowledge gained from these activities was incorporated in the update of the German standardization roadmap, see [5]. In contrast to the previous versions, in particular a separate chapter on "Use Cases" was added. Figure 1 shows the core picture of this new chapter.

Figure 1 illustrates in particular the conceptual distinction between *business scenarios* and *use cases* described on different level of detail:

- Business-scenarios primarily describe a business context, which is addressed by the business viewpoint. Here the basis is a value-network of business stakeholders and

Figure 1: Proposed structure for use cases (according to [5])



Source: DIN/DKE

each business stakeholder is characterized by his own business model. Relations between the business stakeholders within the value-network are characterized by value propositions[1].

- Use cases primarily describe the interaction of technical stakeholders (later called "roles" in this paper) with a technical system. They are addressed by the usage viewpoint. Thus, use cases describe the context of a technical system and high-level requirements, how the technical system interacts with the context. Use cases can be described on different level of detail. Figure 1 illustrates the possibilities to use for the description with the IIRA template or the more detailed template of IEC 62559-2.

A central recommendation of the German standardization roadmap in the context of "Use Cases" is to create further use case descriptions and to classify them using the structure as shown in Figure 1. Various international activities in the context of use cases, such as cooperation's between Germany and Japan resp. China, also integrate their activities into this overarching structure.

Objectives

Following the good experiences and the positive feedback after having created the use case descriptions, the working group "Modeling Example" of the VDI/VDE-GMA Technical Committee 7.21 "Industrie 4.0 – Terminology, Reference Models, and Architectural Concepts" decided to develop another example for a usage view following the proven methodology. The working group was guided by the observation that there is an opportunity to fill the current gap

1 Besides business value there may be technical value and administration value also. If such technical or administration value is important from a business perspective, this would be articulated as a business value in an explicit way.

between the scenarios resp. use cases discussions and the technical discussions, especially in the working group "Reference architectures, standards and norms" of Plattform Industrie 4.0.

The working group "Modeling Example" want to link the more technical discussions of the asset administration shell in the context of Plattform Industrie 4.0 with an application perspective, which identifies the stakeholders of an asset administration shell and their concerns. These concerns frame the *high-level requirements* of an asset administration shell. However, the strategic purpose of the asset administration shell should be elaborated in a more explicit way, too, or to paraphrase this by explaining the business value of the technical and administration value generated by the asset administration shell. This paper does not claim to answer all these questions, but wants to share ideas with regard to the *purpose* of the asset administration shell (see chapter "Asset-oriented Information and Function Structuring") and to describe a *usage view*, i.e. a "black-box" description, to formulate the *high-level requirements* for an asset administration shell on a specific balanced level of detail (see chapter "Relation to Application Scenario Value-based Service" this "black box" description of the asset administration shell is linked to the application scenario Value-based Service, see [3] and [4]. This was also done to explain the generic concepts described in this paper by some more concrete examples. It is planned to link the "black box" description of the asset administration shell also to other application scenarios as soon as these application scenarios are described on a similar level of detail as the application scenario Value-based Service.

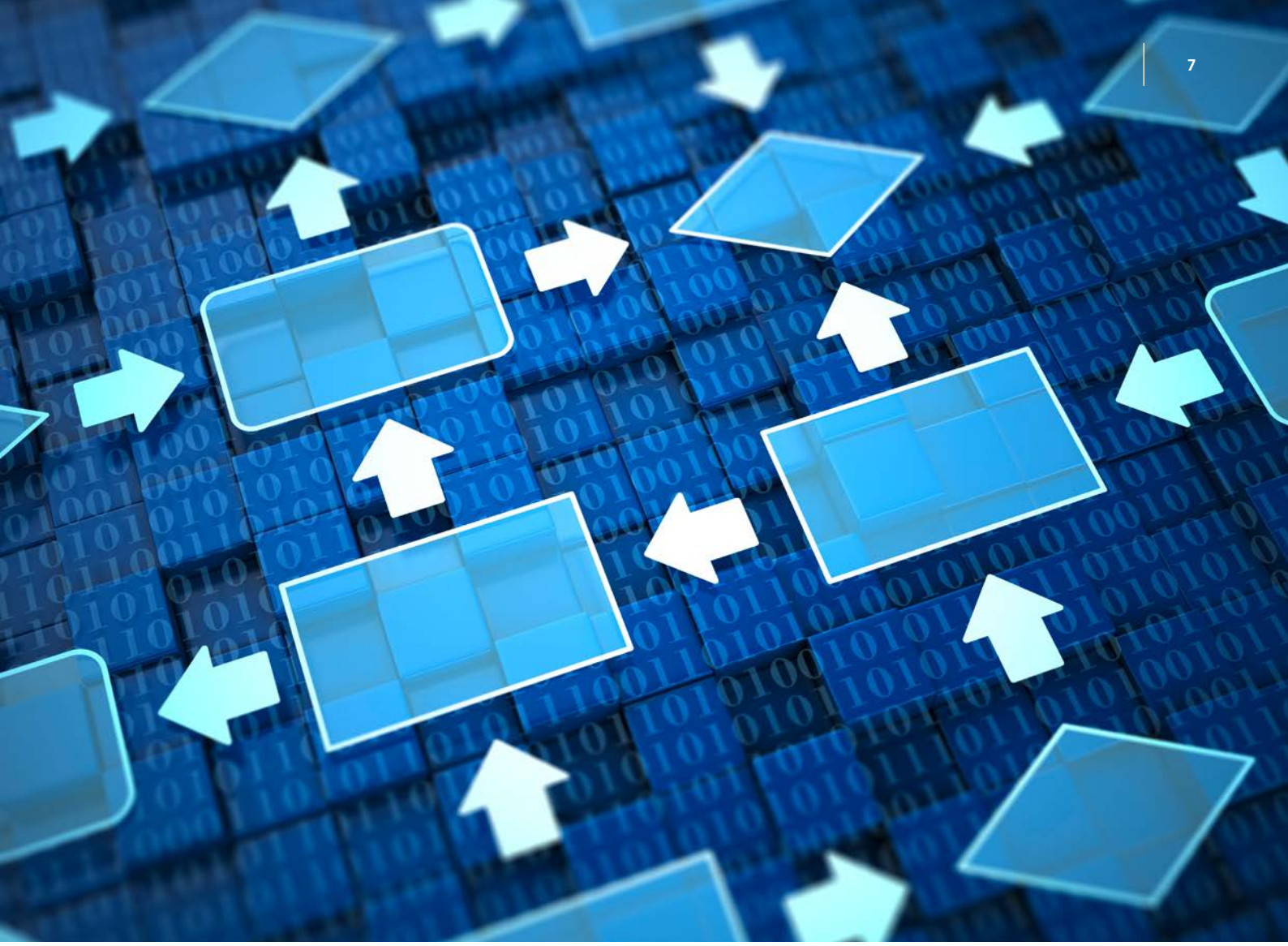
In addition, it should be ensured that this "black box" description is consistent with the many existing, more detailed descriptions of the asset administration shell, see chapter "Annex: Relation to Other Activities and Publications".

As with the previous activities, the goal was to find a suitable level of abstraction for the description of the usage view under the following boundary conditions:

- Understandable for persons outside of the author team
- Completeness (in the sense of an "80% rule") with respect to the concept of asset administration shell
- Manageable size (about 20 pages for the description of the usage view)

The target audience of this document is *users* who want to better understand the benefits and applications of the asset administration shell. Our understanding of "user" is a solution and process architect having the interest to understand a technical concept in an application context, for more details see chapter "Annex: Relation to Other Activities and Publications". In addition, system and software architects and even technical implementers are addressed in the sense to understand the high-level requirements, but not to have guidance for design and implementation concepts.

The first draft of the working group "Modeling Example" was then intensively discussed and refined together with the "Use Case Task Force" in the International Standardization Action Group, Robot Revolution & Industrial IoT Initiative, so that these results are now available as a joint publication. These activities are part of the Germany-Japan cooperation within IEC TC65 Smart Manufacturing and were moderated by the Standardization Council Industrie 4.0.



Asset-oriented Information and Function Structuring

History

To better understand the overall objective of this paper, a brief review of the past work of this working group is necessary. Figure 2 shows some selected publications.

Overall, a top-down approach was followed by first preparing possible business views for the application scenario Value-based Service. Thereafter, the usage view was considered and elaborated successively on different level of detail. In this respect, it would be logical to develop a functional view as the next step for the application scenario Value-based Service. Figure 3 illustrates an example of what this would mean in terms of content.

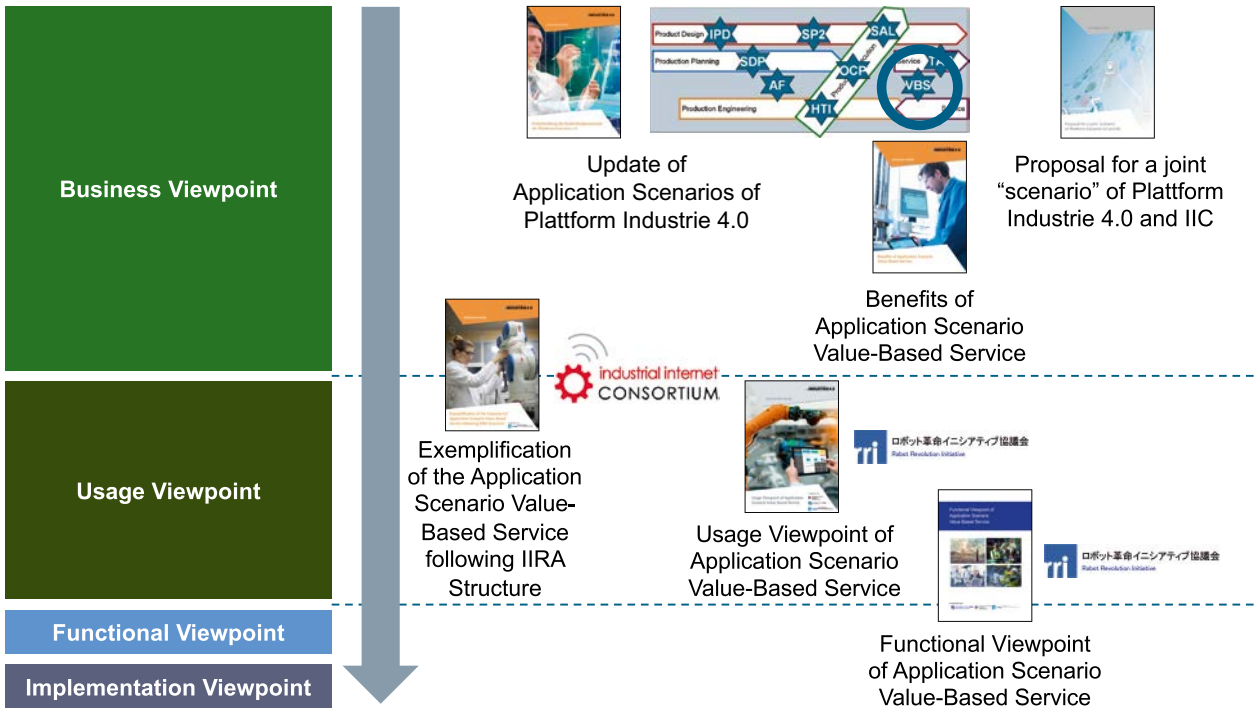
In detail, the functions of an IIoT system (according to [7]) would be structured and described according to the functional domains proposed by the Industrial Internet Consortium. In particular, one would specify the interrelations and structure of functional components, the interfaces and interactions between functional components and the relation and interactions of the system with external elements in the environment. In addition, one would consider a suitable information model and important cross-cutting functions and system characteristics.

The working group “Modeling Example”, however, has decided not to work on this, but to pursue a “complementary” approach.

Observations from Manufacturing Industries

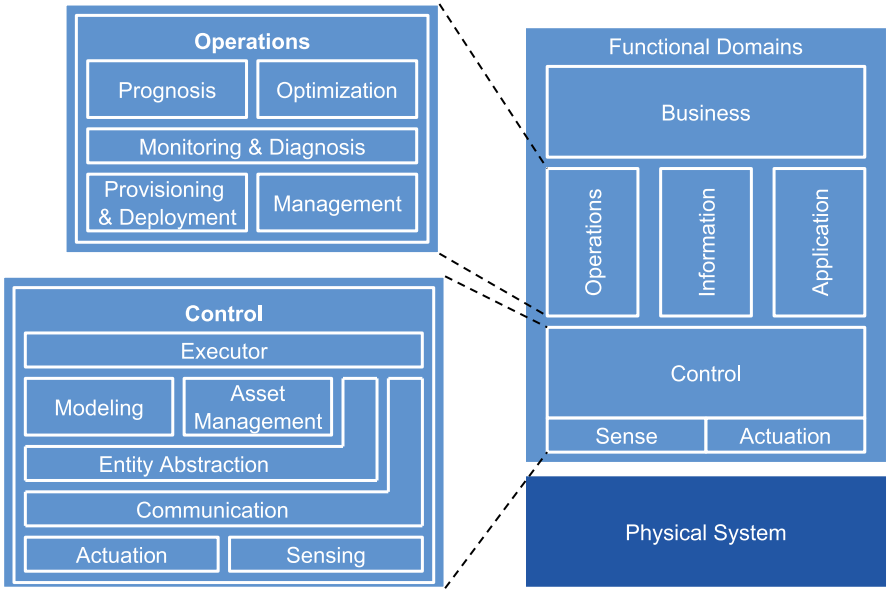
Analyzing typical information technology applications in the manufacturing environment, one realizes that many of the information and functions can be associated with

Figure 2: Selected publications in the context of the application scenario Value-based Service



Source: Plattform Industrie 4.0

Figure 3: Functional Viewpoint of Industrial Internet Architecture (according to [7])



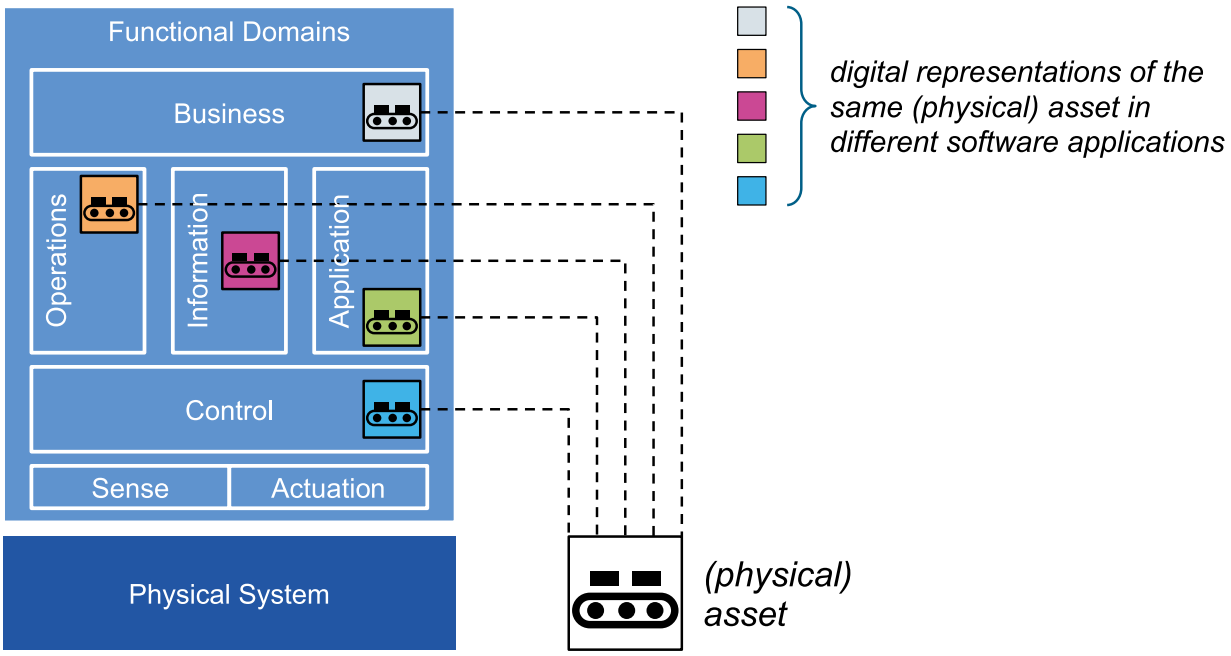
Source: Plattform Industrie 4.0

so-called assets, see [8]. The assets have intrinsic properties and the various software applications internally map these properties for their own purposes in the sense that during the design of the software application these properties have to be modeled suitably. Since every software application pursues different purposes, the mapping resp. modeling of the intrinsic properties of an asset depends on the software application under consideration. Figure 4 exemplifies this with references to the functional viewpoints proposed by the Industrial Internet Consortium.

These specific mappings are designed by individual application developers. Today, this is typically done in the minds of these developers.

Now, especially in the manufacturing industry, these assets itself are subject to modification. Typical examples are physical changes, reconfigurations or usage changes because of optimization of processes or products. These changes must be suitably considered with respect to the individual software applications and their usage. In the case of a physical change for example it could be necessary to modify some CAD- or wiring-diagrams manually using the corresponding software applications and in addition in some software applications for process control some parameters have to be adjusted. In the case of a reconfiguration for example some synchronized changes in various software applications for process control and maintenance have to be executed. This change process is complex, since there is a variety of software applications, and highly prone to errors, since these changes usually need to be done by hand, for example even by writing new source code or by installing additional software. To make things worse, the nature and number of such software applications is increasing due to the increasing IT penetration of the manufacturing industry and thus the challenges of the described information management are increasing.

Figure 4: Various representation of the same asset in different software applications



Source: Plattform Industrie 4.0

In addition, there is a further trend that information about and functions of an asset will increasingly be shared by different business stakeholders, as described in the business view of the application scenario value-based service, see [2]. Regarding this trend, the amount of information to be exchanged increases as well as the number of functions to be shared increases.

To counter these trends, the concept of *asset administration shell* was created. The idea of the asset administration shell is to structure the information and functions in the context of the manufacturing industry based on the assets in a uniform manner. This allows the information about and functions of an asset to be decoupled from the application specific interpretation by a software application. In a sense, this contributes to an improved *interoperability*[2] of assets from an information technology perspective. A unified, asset-oriented information structure helps to better master the complexity of information management in manufacturing industries by both reducing the effort and increasing the quality of information. Of course, this requires a certain effort in asset-oriented information structuring in advance. For the individual user, this preliminary investment is reduced to the extent that standardization bodies address this topic and the corresponding standards are accepted in the market.

Thus, the general context of the asset administration shell can be summarized in the following way:

- It is a good practice that in the industrial environment[3] a user structures the required information and functions according to the assets and their structure.
- The information and functions structured in this way should be managed throughout the life of the asset logically at one location.

In order to support this approach, the asset administration shell should provide the following capabilities:

- Mechanism how to access to information and functions of an asset in a uniform way
- Reference to definitions of semantics of information and functions
- Definition of visibility of and access to information and functions with respect to other stakeholders
- Flexible deployment with respect to a computing infrastructure[4]

These capabilities have to be realized by appropriate information technology concepts.

Overall Roadmap

The concept of asset administration shell is basically not a new idea. Figure 5 illustrates how this idea has developed evolutionarily over the last years. In addition, this concept is influenced by major technology developments outside of the manufacturing industry. These are additional drivers affecting the overall roadmap.

To focus the discussion on the asset administration shell, it is expedient to distinguish the following evolutionary stages:

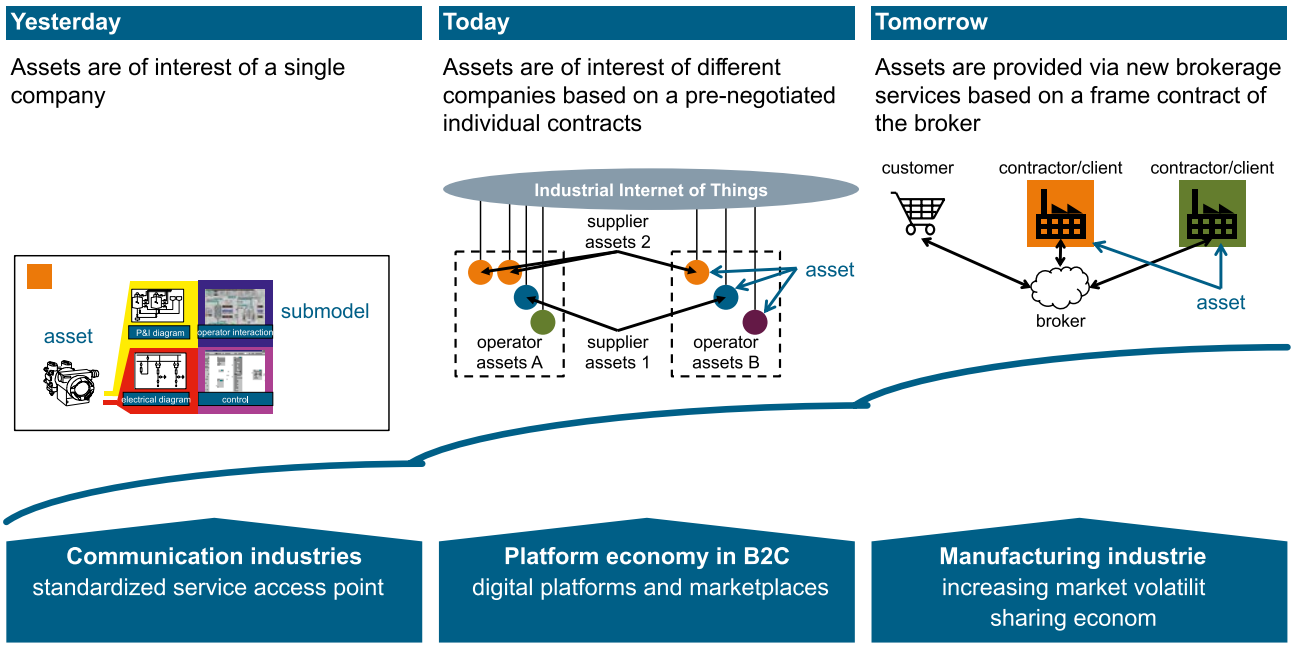
- The principles of asset- (respective object-) oriented information modeling are known since more than 20 years and tooling support is available since more than 15 years. In the past many “similar” approaches were propagated, e.g. ABB Aspect Object, Comos Master Object, etc. In the past, the application was primarily restricted to the usage of assets from the perspective of a single company. Although these principles were often applied in practice, especially in process industries, they are still not used across all industries today. Consistent implementations are often very challenging.

2 In this context interoperability requires two measures: a common understanding of the mechanism, how to access information and functions in a uniform way, and a common understanding of the concrete content. Depending on the application, this requires cross-company standardization.

3 Note that in our understanding industrial environment is broader than an IIoT system in the sense of the Industrial Internet Consortium. For example, asset-oriented information structuring can be applied to the usage of a CAD-engineering tool system, too. In addition, structuring information and functions according to assets is not restricted to manufacturing industries.

4 The topics of flexible deployment and computing infrastructure are not considered in detail in this document. Therefore a separate activity was started in the VDI/VDE-GMA Technical Committee 7.21.

Figure 5: Overall roadmap evolution of idea of asset administration shell



Source: Plattform Industrie 4.0

- In the current discussion about the role of IIoT platforms – see especially the application scenario Value-based Service, see [1] – the focus of the application of the asset administration shell changes. Typically, assets are installed all over the world and connected via an IIoT platform. There are different (business) stakeholders having an interest in the assets during their lifetime, e.g. the operating company of an asset and the supplier of an asset. These stakeholders sign individual bilateral legal contracts on their mutual interest in an asset.

- However, the current discussions often also address future application scenarios such as the application scenario Order-Controlled Production, see [6]. Here assets are published and offer their capabilities via a brokerage service to be used by customers. A customer requesting asset capabilities negotiates with the brokerage services based on a frame contract. This frame contract is defined by the brokerage service.

This roadmap is embedded into the following technology and market drivers:

- In the past, the communications industry has been very successful in establishing a standard with the ISO/OSI seven-layer model, which helped to bridge the requirements and solutions in this market and to come to a common understanding. By standardizing the various layers and transitions between these layers by the so-called service access points and associated communication protocols, interoperability among the various companies on a technical level was achieved. The underlying ideas as well as technical concepts are a major source of inspiration for the current discussion about the asset administration shell.

- Today, there are several digital platforms and marketplaces in the business-to-consumer market, e.g. Amazon,

AirBnB, eBay. This phenomenon is called platform economy. These developments are projected onto the Industrial Internet of Things and thus additionally influence the discussion about the opportunities and effects of an asset administration shell.

- Manufacturing is changing faster than ever and driven by ensuring the required quality, increasing efficiency, shortening time-to-market, and enhancing flexibility. The increasing volatility of the market is the key driver for future scenarios in a sharing economy that generate additional high-level requirements for an asset administration shell.

Guiding Principles for Underlying High-Level Requirements Engineering

As illustrated especially in Figure 5, many usage scenarios and applications of the asset administration shell are possible. In order to focus the discussion of the high-level requirements of the asset administration shell, we applied the following overall guiding principles in our considerations:

- We focus on a purely information technology perspective: An asset provides *asset services* and these services represent software functionality (and not capabilities in the real world). Of course, the execution of software functionality can have effects beyond the information world.

- Our perspective is from the *asset* of a stakeholder: Such an asset has a *value* for the stakeholder. This is the case if the stakeholder is for example the owner of a physical asset, but it is also the case if the stakeholder is not the owner of a physical asset but has information about or functions for that asset. Therefore, the stakeholder of the asset alone wishes to decide, whether and to whom to inform about the asset of one's own interest and the related information and functions about the asset respectively. This provides a basic mechanism to *protect intellectual property* about the asset.

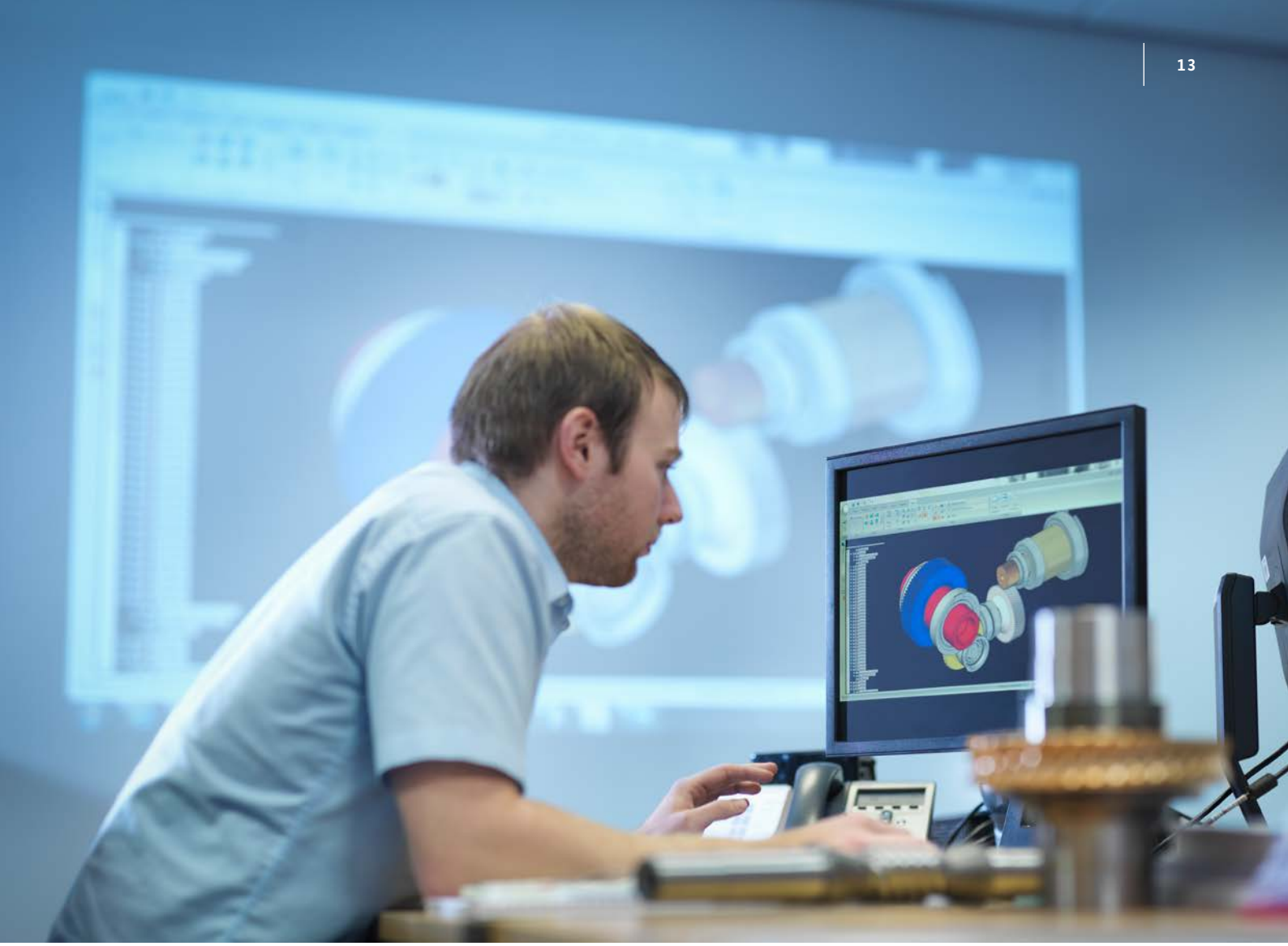
- In principle, there are the following sharing mechanisms to other stakeholders:

  - The stakeholder of an asset can make some of the information and functions, which are under the own control, available to another stakeholder, e.g. the provision of an asset lifetime prognosis, where some stakeholder of the asset calculates this prognosis and provides it to other stakeholders.

  - A special case is that information or functions of a physical asset is stored or executed on a computing resource of the asset itself, e.g. the provision of energy consumption of a physical asset, where the owner of the physical asset calculates the consumption on the asset itself and provides it to other stakeholders.

  - A stakeholder of an asset provides asset relations, i.e. context information of the asset, to another stakeholder for exploration purposes, e.g. the model series ("type" asset) of an asset ("instance" asset), where the supplier of an asset provides information about the model series to the user of the asset.

- Possible new brokerage roles require a high degree of semantic standardization. Therefore we do not discuss brokerage roles in detail, but we refer to possible extensions, which are necessary to support new brokerage services.



Usage View of Asset Administration Shell

To understand the Usage View, we first introduce concepts of the usage view*point* proposed by the Industrial Internet Consortium, for details see [7]:

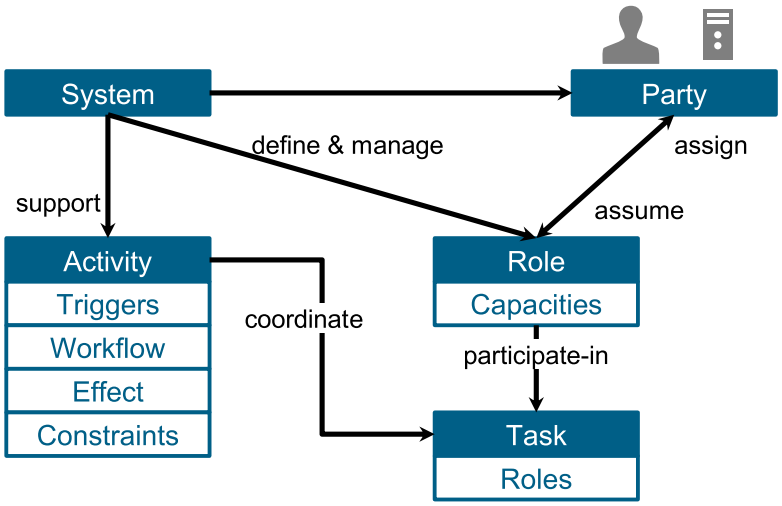
- The basic unit of work is a *task*[5]. A task is **carried**[6] out by a party assuming a role.
- A *role* is a set of capacities assumed by an entity to initiate and participate in the **execution** of, or consume the outcome of, some tasks or functions in a system as required by an activity. Roles are assumed by parties.
- A *party* is an agent, human or automated, that has autonomy, interest and responsibility in the **execution** of tasks. A party **executes** a task by assuming a role that has the right capacities for the execution of the task.

  A party may assume more than one role, and a role may be fulfilled by more than one party.
- An *activity* is a specified coordination of tasks required to realize a well-defined usage or process of a system. An activity has the following elements:
  - A *trigger* is one or more condition(s) under which the activity is initiated.
  - A *workflow* consists of a sequential, parallel, conditional, iterative organization of tasks.
  - An *effect* is the difference in the state of the system after successful completion of an activity.

5 The tasks according to [7] include a Functional Map referring to the Functional Viewpoint and an Implementation Map to the implementation Viewpoint. Since we are focusing on the Usage Viewpoint only, we do not consider Functional resp. Implementation Maps.

6 The **bold** marked terms refine and illustrate the single term “participate-in” in Figure 6.

Figure 6: Overview of Usage Viewpoint (according to [7])



Source: IIC

- *Constraints* are system characteristics that must be preserved during execution and after the new state is achieved.

At this point we would like to point out the difference regarding the terms "viewpoint" and "view"[7]: An (architecture) view expresses the architecture of a system from the perspective of specific system concerns, whereas an (architecture) viewpoint establishes the conventions for the construction, interpretation and use of architecture views to frame specific system concerns. For more details we refer to [7] or even ISO/IEC/IEEE 42010.

Overview

Starting with a very high-level overview, we see the following main characteristics of the asset administration shell. We use the term *asset service* as a uniform concept to access information and functions of an asset:

- The asset administration shell comprises an *asset service registry* of an asset from the perspective of an organization having an interest in the asset. Such an asset service registry declares all asset services of the asset, which are of interest for the organization.
- The asset administration shell provides secure access to asset services.
- Asset services can be standardized by standardization organizations.
- The asset administration shell offers the possibility for structuring including encapsulation following the relationships between assets.
- The asset administration shell offers different ways of deployment and implementation[8], especially depending on the capabilities of the asset.

Figure 7 shows an overview of the usage view of the asset administration shell. In gray, the system under consideration is shown and in dark purple the roles that are outside of the system boundary of the asset administration shell. These roles interact with the asset administration shell. The

7 We must admit that in older publications, especially in [2], [3], and [4], we mistakenly have not made this distinction. We have spoken of the business resp. usage resp. functional *viewpoint* of the application scenario Value-based Service, but we discussed business resp. usage resp. functional *views*.

8 Especially asset service will evolve over the time. The underlying computing infrastructure has to provide appropriate capabilities to manage this evolution over the time, see chapter "Computing Infrastructure".

color-filled rectangles indicate belonging to an organization, which has an interest in the asset under consideration.

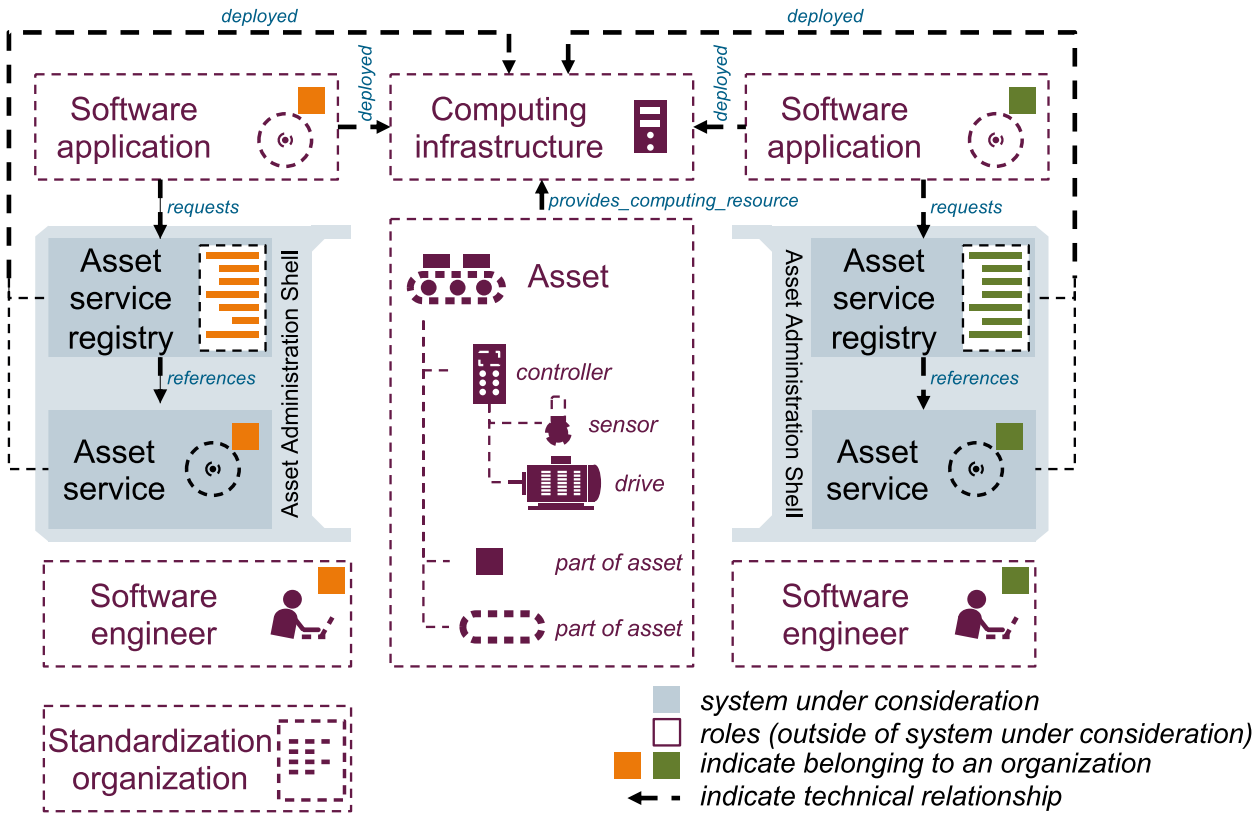
The asset administration shell as *system under consideration* conceptually includes of the following components:

- Asset service registry: An asset service registry belongs to an organization. The scope of the asset service registry is determined by the associated asset *and* the organization considered.
- Asset service: This is a software functionality, which is provided to software applications via an asset service registry. Asset services belong to an organization.

The following *roles* are to be considered in the context of the asset administration shell:

- Asset: Assets are entities with a value for an organization.
- Software engineer: This role belongs to an organization with interest in the asset under consideration. The software engineer is responsible to implement the interest of the organization in the asset by an asset service registry, asset services and software applications. The asset administration shell is a technical concept from an information technology perspective; therefore, information technology capabilities are requested.

Figure 7: Overview of usage view of asset administration shell



Source: Plattform Industrie 4.0

- Software application: This is a software program (often also called client) that uses the asset services provided via an asset service registry. Software applications include the purpose for which an asset administration shell has been created[9]. Software applications belong to an organization.

- Computing infrastructure: The computing infrastructure is necessary to deploy and execute implementations of the asset service registries, asset services, and software applications. Note that computing resources can also be provided by an asset.

- Standardization organization: This is an organization with the objective to standardize certain asset services.

Example for Illustration

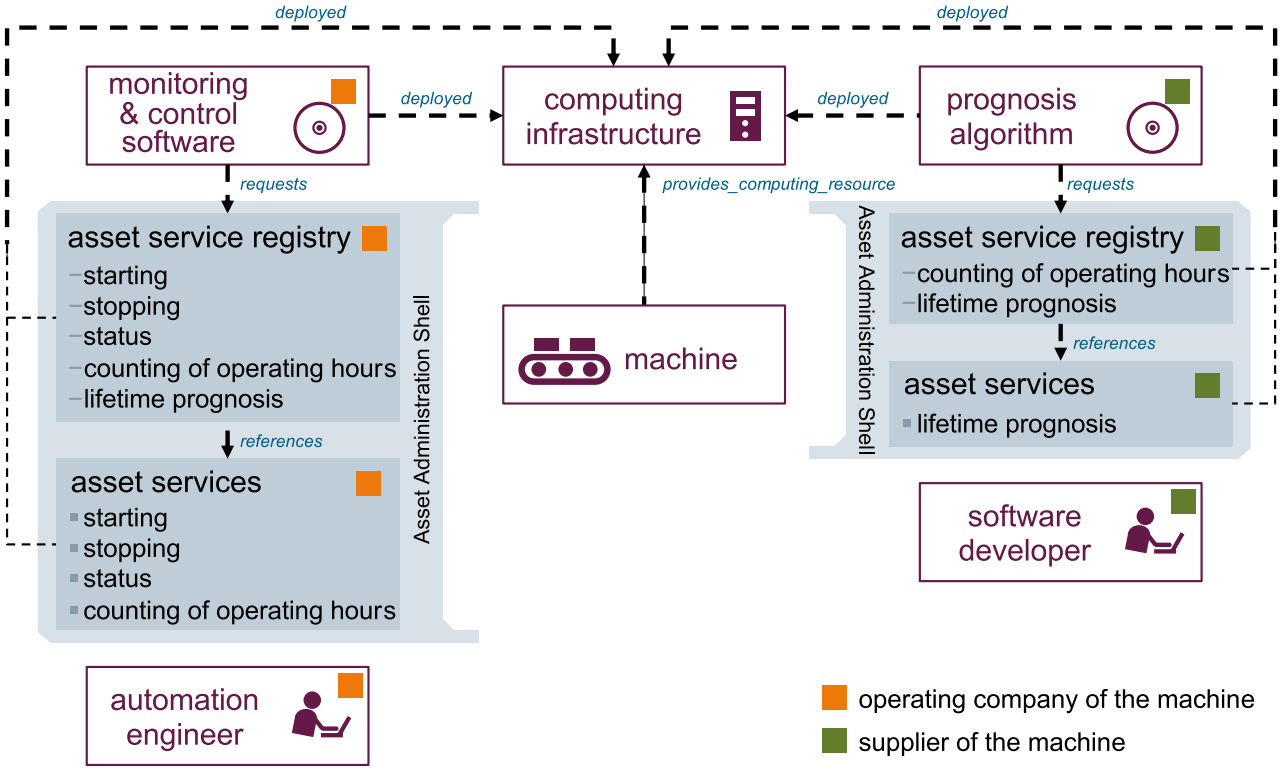
We use the usage view of the application scenario Value-based Service, see [3], to illustrate the main concepts of the asset administration shell according to Figure 8.

- We consider two organizations: the operating company of the machine indicated in "orange" and the supplier of the machine indicated in "green".

- As *asset* we consider a machine (named "connected asset" in [3]).

- As "orange" software engineer we consider an automation engineer. The automation engineer belongs to the

Figure 8: Example for illustration



Source: Plattform Industrie 4.0

9 In our approach, software applications are not shared with other organizations; if this is requested, the software application should be encapsulated by an appropriate asset service.

organization which operates the machine, and integrates the automation of the machine into the overall automation system of the plant. This includes the implementation of capabilities requested to operate the machine by a human operator.

- As "green" software engineer we consider a software developer. The software developer belongs to the organization which supplies the machine, and implements the various capabilities of the machine realized by software. The requirements typically result from the product management of the supplier of the machine and – in the case of non-standard machines – also from the customer of the machine.

- Asset services of the operating company of the machine are e.g. starting, stopping, status (on, off, out-of-operation, fault, etc.) and counting of operating hours. The counting of operating hours comprises an aggregated value and more detailed information like e.g. the load profiles during operation of the machine.

- An asset service of the supplier of the machine is a lifetime prognosis, which can be extended to provide condition-based maintenance support. To calculate this information the supplier of the machine needs access to the detailed information of the counting of operating hours.

- The asset service registry of the operating company of the machine references the services starting, stopping, status, counting of operating hours (all provided by the operating company of the machine) and lifetime prognosis (provided by the supplier of the machine).

- The asset service registry of the supplier of the machine references the services lifetime prognosis (provided by the supplier of the machine) and counting of operating hours (provided by the operating company of the machine).

- A software application of the operating company of the machine is e.g. the customer specific monitoring & control software of the machine.

- A software application of the supplier of the machine is e.g. a generic prognosis algorithm, which prognoses the lifetime of the machine based on experiences gained by the supplier of the machine over the years from the usage of his machines installed all over the world at different operating companies of the machines.

- The starting, stopping, status and aggregated counting of operating hours could be standardized by various suppliers offering similar machines. The detailed information of the counting of operating hours will not be standardized, because this may be a competitive differentiating feature.

System under Consideration

In this section, we describe the system under consideration in more detail. There are some forward references in this explanation because we have decided to describe the system under consideration and the roles independently.

Asset Service

A service (according to Wikipedia[10]) is software functionality or a set of software functionalities with a purpose that different software applications (so-called clients) can reuse the functionality for different purposes.

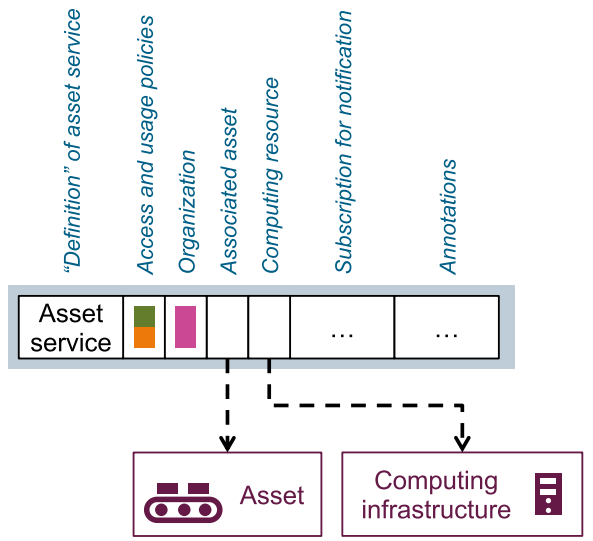
Figure 9 illustrates the concept of asset service:

- An asset service is a service associated to an asset and belongs to an organization. Some software engineer from the organization is responsible to integrate an asset service into the computing infrastructure following all requested policies, e.g. with respect to IT security.

- The use of an asset service is controlled with appropriate policies, e.g. based on the "identity" of the software application requesting the asset service. The access and usage policies of an asset service are defined by some software engineer from the organization of the asset service.

10 There are many different definitions for "service" including ISO, IEC, etc. We think that the definition according to Wikipedia supports the purpose of this paper very well.

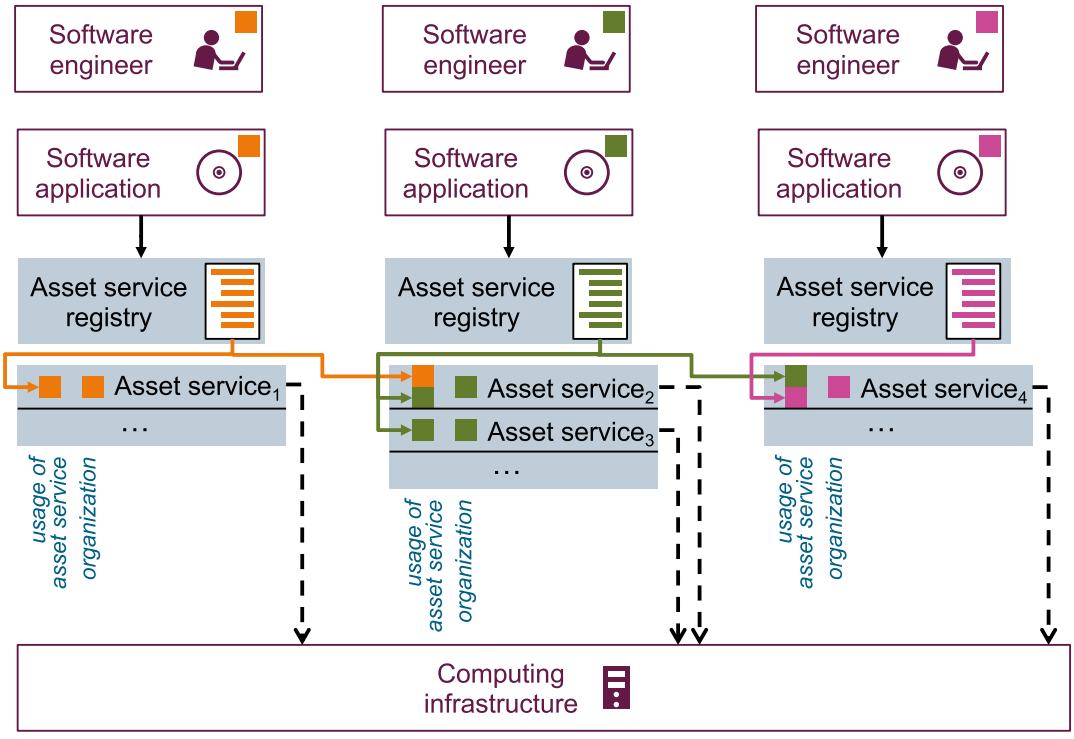
Figure 9: Concept of asset service



Source: Plattform Industrie 4.0

- An asset service $service_N$ of an asset $asset_n$ can call (as a client) an asset service $service_M$ of another asset $asset_m$. To guarantee well-structured programming there should be a relation between the assets $asset_n$ and $asset_m$. Note that the concept of relation between assets is explained in more detail in the section "Asset".
- Each asset service includes a reference to a computing resource of the computing infrastructure on which the asset service is deployed.
- An asset service or software application can subscribe itself to be notified in the case of change[11] of an asset service. Notifications have to be granted by some software engineer from the organization of the asset service.
- Asset services can be annotated, for example with regard to standardization.

Figure 10: Illustration of concept of asset service registry



Source: Plattform Industrie 4.0

11 Change in the sense of change of functionality.

Asset Service Registry

A core concept of the asset administration shell is the so-called asset service registry as illustrated in Figure 10.

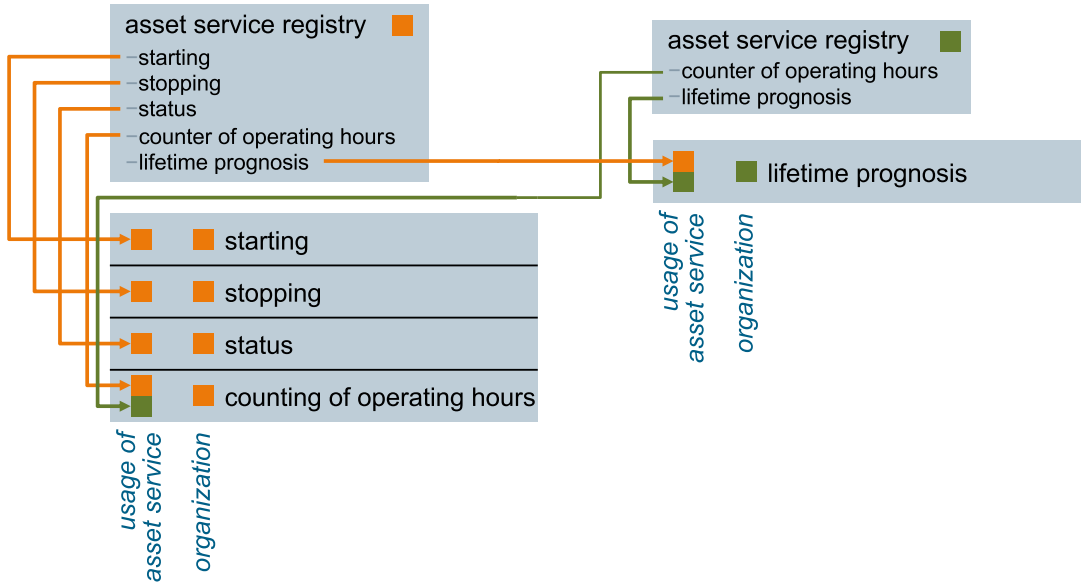
An asset service registry belongs to an organization having an interest in the asset. It defines from the perspective of such an organization all asset services relevant for this organization:

- Following the ideas of asset-oriented information and function structuring, there is from the perspective of an organization exactly one asset service registry for an asset[12].
- Two different organizations may use the same asset service.
- Software engineers belonging to different organizations can implement the same asset service, i.e. $service_N$ of $organization_n$ and $service_M$ of $organization_m$ may have the same semantics. This is for example the case if $organization_n$ as one company does not want to depend on $organization_m$ as another company and therefore implements its own $service_N$ even though $organization_m$ provides $service_M$. In this situation a software engineer belonging to $organization_n$ would use $service_N$ and a software engineer of $organization_m$ would use $service_M$.
- If any software application as a client wants to use an asset service, this service has to be addressed in an indirect way via the asset service registry.
- The management of access and user rights can be defined even more detailed, e.g. different roles within the organization or distinction between e.g. discovery, read-only, and write access rights of software applications, etc.

Figure 11 illustrates the concept of asset service as described in Figure 10 based on the example as shown in Figure 8.

There may be various *relations* between assets. As illustrated in Figure 12 such relations may result in counterpart relations between corresponding asset service registries. Note

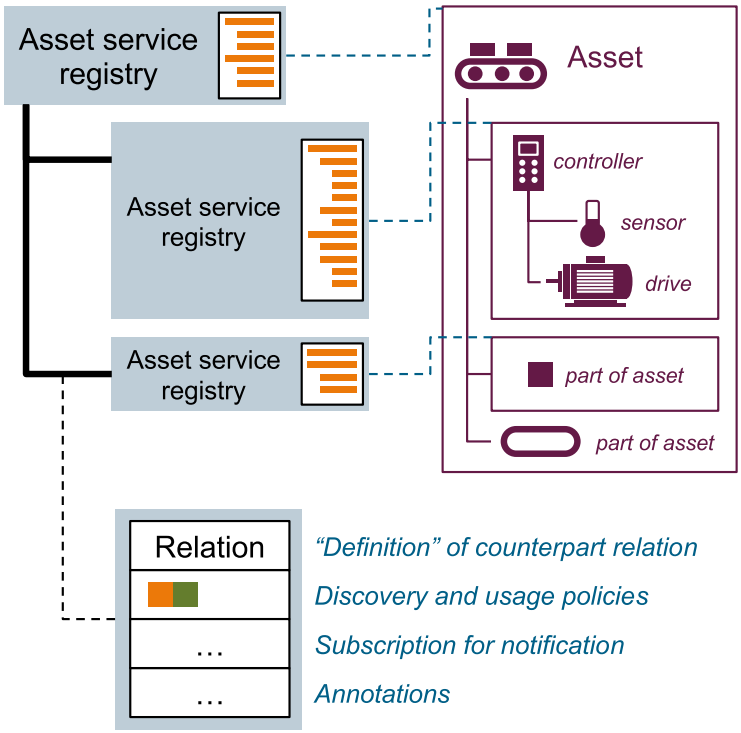
Figure 11: Illustration of asset service registry and asset services



Source: Plattform Industrie 4.0

12 The overall concept is not hurt if there are multiple asset service registries for an asset in one organization, but this would conflict with the concept of asset-oriented information and function structuring.

Figure 12: Illustration of concept of relations between asset service registries



Source: Plattform Industrie 4.0

that the concept of relation between assets is explained in more detail in section "Asset":

- In the case of a relation[13] between $asset_n$ and $asset_m$ and if this relation is relevant for the organization having an interest in $asset_n$ some software engineer belonging to this organization defines a counterpart relation between the asset service registries of $asset_n$ and $asset_m$.
- The software engineer can define discovery and usage policies of the counterpart relation between the asset service registries for other organizations. These policies control the access to the counterpart relation by asset services and software applications.
- An asset service or software application can subscribe itself to be notified in the case of change of a counterpart relation. Notifications have to be granted by the software engineer.
- Counterpart relations between asset service registries can be annotated, for example with regard to standardization.

Roles

In this section, we describe the various roles in more detail. The forward references established in the section "System under Consideration" will be picked up here again.

Asset

An *asset* is an item which has a value for an organization and which is administrated individually for this reason.

A physical world asset is *owned* by some organization; the owner may change over the lifecycle of an asset. During a transfer of ownership various information about the asset

13 With respect to modeling, a relation between $asset_n$ and $asset_m$ is an independent model object relating the model objects of $asset_n$ and $asset_m$.

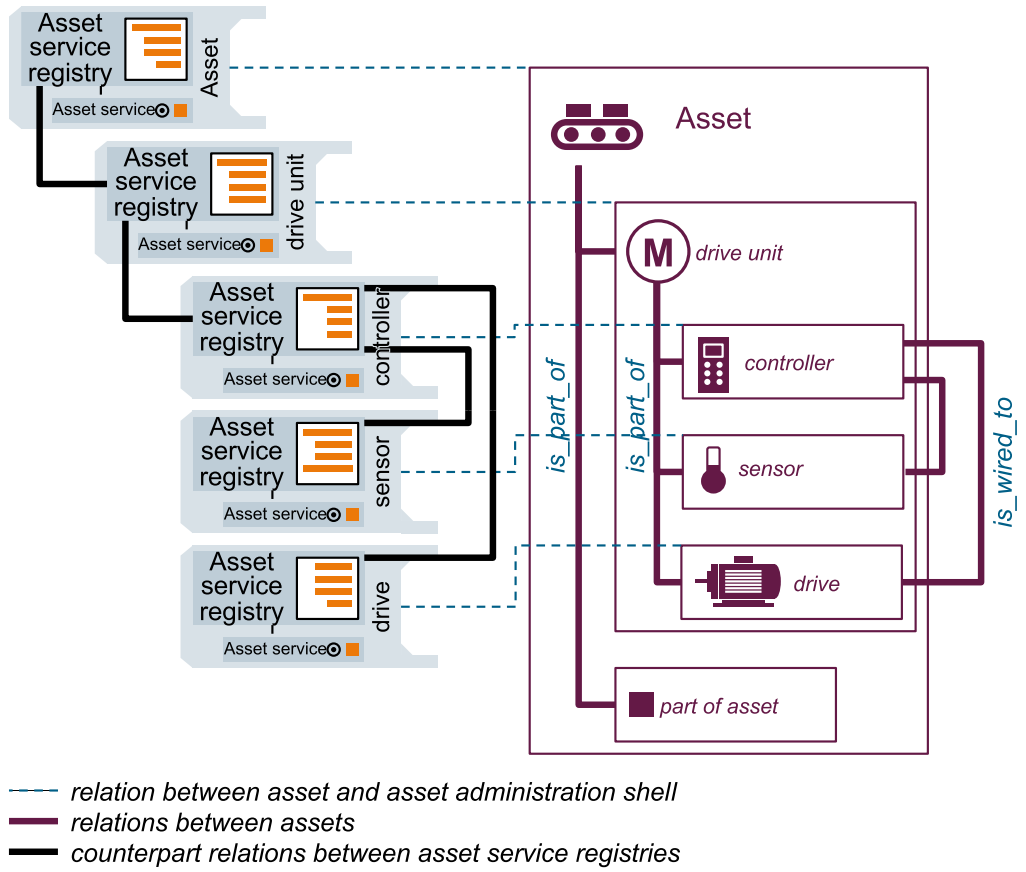
may be handed over, nevertheless, the organization of the new owner is responsible to create an own asset service registry and asset services and to integrate these suitably into the computing infrastructure. For more details see section “Activities”.

A physical world asset may comprise own computing resources and implementations of specific asset services. These asset services belong to the organization of the owner of the asset. It is in the responsibility of some software engineer of this organization to integrate the computing resources and implementations suitably into the computing infrastructure.

Relevant relations between assets[14] (from the perspective of a specific organization) have counterparts in form of relations between the corresponding asset service registries. Such relations could be regarded as specific standardized asset services, but we distinguish between relations and asset services. We illustrate such relations between assets and possible counterpart relations between asset service registries based on examples for a physical world asset and a virtual world asset:

- Figure 13 shows an example for relations between assets in the physical world and illustrates counterpart relations between the corresponding asset service registries.

Figure 13: Illustration of relations of a physical world asset (example machine)



Source: Plattform Industrie 4.0

14 Our concept is based on a systems engineering approach: systems can be structured hierarchically in terms of an *is_part_of*-relation between a system and its subsystems, and various connection-relations may exist between the subsystems, e.g. *is_wired_to, communicates_with, feeds_material_to*. We assume such a generic view of a system. In the context of the asset administration shell, the system is the set of all assets considered, and there exist relations between these assets in the form of hierarchical and connection relations.

The right part in Figure 13 shows a machine consisting of a drive unit and other parts. The drive unit, in turn, consists of a controller, a sensor and a drive. These relationships are *is_part_of*-relations, i.e. a component consists of subcomponents. In addition, the sensor and the drive are communicatively connected with respect to their communication capabilities to the controller. These are *is_wired_to*-relations, i.e. a component can communicate with another component.

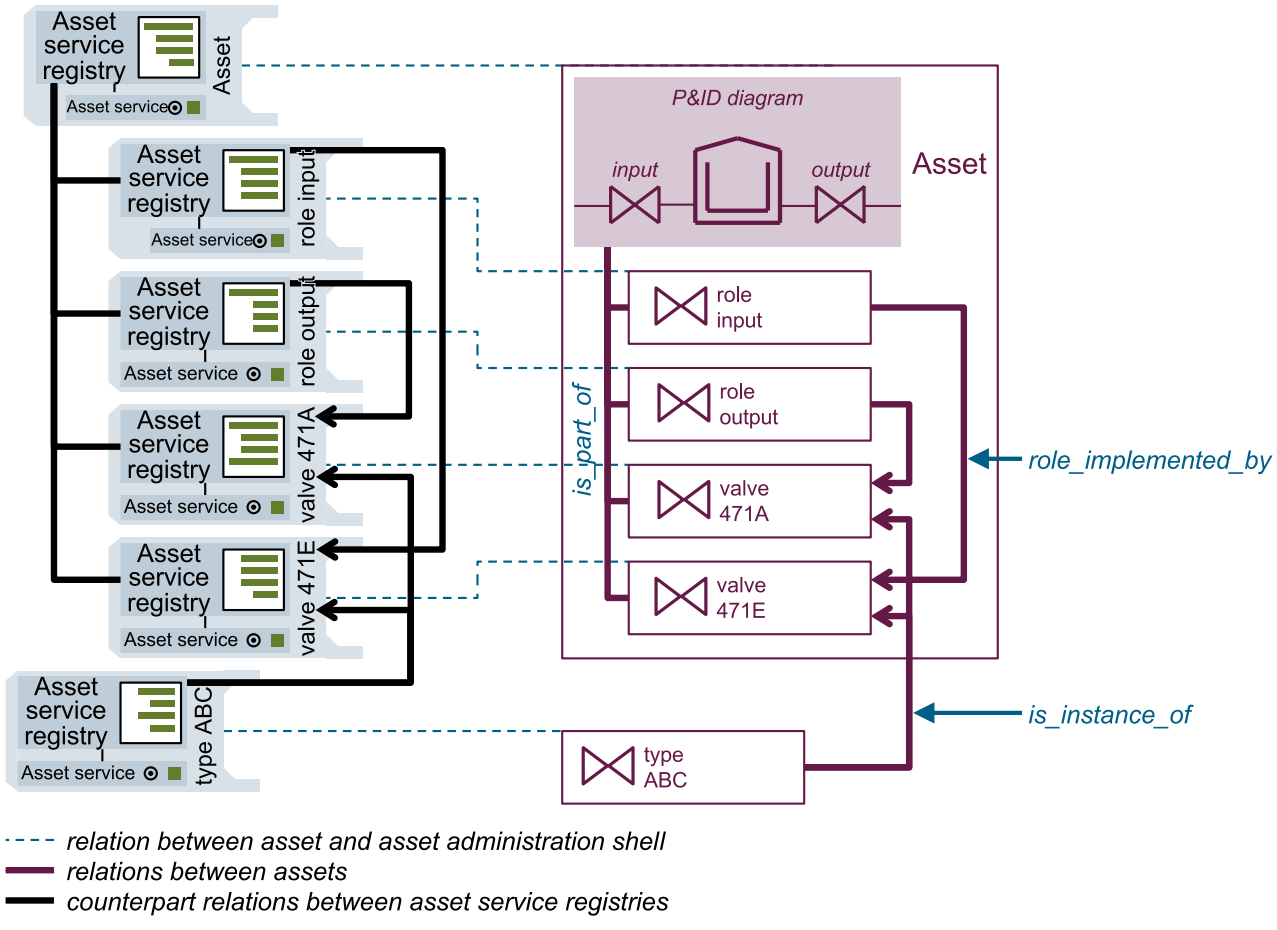
Due to a design decision, the software engineer decided to model the machine (i.e. asset in Figure 13), the drive unit, the controller, and the sensor as assets with appropriate asset administration shells. In addition, the software engineer decided not to regard the other parts of the machine as assets. Note also that (because of a design decision) the *is_part_of*-relations between drive unit and sensor resp. drive are not modeled in the form of counterpart relations between the corresponding asset service registries.

It should be noted that not all of the assets of the physical world shown in Figure 13 necessarily have to belong to the same owner.

- Figure 14 shows an example for relations between assets in the digital world and illustrates counterpart relations between the corresponding asset service registries.

The right part of Figure 14 shows a P&ID diagram in which, inter alia, two roles (i.e. input and output) and two implementations of these roles by valves (i.e. valve 471A and

Figure 14: Illustration of relations of a virtual world asset (example piping & instrumentation (P&ID) diagram)



Source: Plattform Industrie 4.0

valve 471E) are represented. The two roles and the two implementations are modeled here as *is_part_of*-relations, and the relation between the concrete roles and corresponding implementations is modeled by a *role_implemented_by*-relation. In addition, it is modeled via an *is_instance_of*-relation that the two valves valve 471A and valve 471E belong to the same series (i.e. type) ABC[15]. The concept of such modeling is usually strongly influenced by the object model from the user's point of view, which is provided by an engineering tool to a user.

The software engineer belonging to the organization having an interest in the P&ID diagram (i.e., asset in Figure 14) has decided to model all these assets as assets with appropriate asset administration shells, and to model all of the relationships between the assets in the form of counterpart-relations between asset service registries.

Software Engineer

A software engineer belongs to an organization that has an interest in the asset under consideration, therefore such a software engineer creates an asset service registry and possibly in addition asset services (which thereby belong to the organization of the software engineer) and software applications (which thereby belong to the organization of the software engineer).

Software Application

A software application is a software program (often also called client) that uses the asset services provided via the asset service registries. In general, software applications represent the specific purpose for an asset administration shell.

A software application belongs to an organization. Some software engineer of this organization is responsible to integrate a software application into the computing infrastructure following all requested policies, e.g. with respect to IT security.

The use of asset services by a software application may be restricted by some software engineer of the organization to which an asset service belongs.

Note that an asset service is a specific kind of software application. Also, a software application can be a specific asset.

Standardization Organization

A standardization organization is an organization with an interest to standardize specific asset services[16] (i.e. their semantics) of a specific set of assets and/or relations between specific sets of assets.

Standardized properties of some type of machines could be for example interaction capabilities (e.g. start, stop), running mode, operating hours, usage profile, and health status. A standardized relational structure of some type of machine could be for example that a reactor consists of an input and output valve and an optional filling sensor.

A specific asset service resp. relation between assets may be subject of different standardization organizations. Standards may be consensus based (like e.g. DIN, IEEE or ISO) or consortia standards (like e.g. OPC-UA).

The industry-specific definitions of machine properties and their implementation in the form of OPC-UA companion specifications, which are currently developed under the guidance of the VDMA, as well as comparable activities of the VDW in the context of the definition of so-called connectors, are examples where concrete standardization is currently taking place in the market. These activities address those subareas of the overall concept of the asset administration shell, in which the various companies currently see the greatest potential for benefits.

15 In the general discussion in the context of RAMI4.0, a type-instance concept was introduced in the so-called lifecycle & value chain axis. In manufacturing industries there are application-dependent different kinds of "type-instance" relationships. For this reason, we do not consider a type-instance concept as an integral part of the asset administration shell, but a type-instance concept could – as illustrated in the example of Figure 14 – be applied application-dependent.

16 Such asset services are typically not limited to specific vertical domains.

Computing Infrastructure

A computing infrastructure provides the following functional capabilities[17]:

- Implementation, deployment and testing capabilities for asset services and software applications: Typically, this is provided via specific software engineering tools supporting the management of entire lifecycle of the software programs.
- Executable asset service registry: This includes the following capabilities
  - Capabilities for configuration of the asset service registry by some software engineer: This includes capabilities to define asset services, to define access rights and notifications, and to define relations between assets. These configuration capabilities are also provided for configuration of software applications and asset services.
  - Capabilities for exploration through software applications and asset services. Examples for such exploration capabilities are the exploration of available assets, of an asset service registry, of related asset service registries and of asset services referenced by an asset service registry.
  - Implementation and execution of asset service registry: It is a design decision whether an asset service registry is implemented by an executable software program or a data structure. But for the sake of interoperability the access to asset service registries should be standardized.
  - Provision of notifications according to notification subscriptions.
- User and access rights management: This includes the following capabilities
  - Controlling and limiting the access of software applications to an asset service registry and the associated asset services.
  - Access of asset service registries of different organizations to asset services of the same asset.
  - Access of asset service registries of $asset_n$ to a related asset service registry of $asset_m$, where $asset_n$ and $asset_m$ are related. The concept of relations is explained in section “Asset”.
- Transparency with respect to asset services whose description and implementation is provided by an asset itself: If some asset provides own computing resources the computing infrastructure offers expandability with respect to these additional computing resources. This includes the consideration of all necessary security policies.
- Communication infrastructure: This includes ensuring confidential and integrity-assured data exchange between asset services, asset service registry and software applications as well as event logging for transparent and traceable interaction between asset services, asset services registry and software applications. If there are requested specific realtime capabilities (e.g. short latencies), the computing infrastructure has to provide suitable capabilities.

There is an *operator* of the computing infrastructure. The operator is responsible to setup, operate and maintain the computing infrastructure. This includes

- to offer a computing infrastructure according to the requested capabilities
- to securely operate the computing infrastructure
- to define the overall security policies (management of organizations, software engineers, integration of asset services, software applications and assets with own computing resources)

A computing infrastructure provides the following non-functional capabilities:

- The computing infrastructure has to provide online capabilities with respect to creation, modification, dele-

17 In this paper we focus on the requirements of a computing infrastructure. It is planned do elaborate within VDI/VDE-GMA Technical Committee 7.21 “Industrie 4.0 – Terminology, Reference Models, and Architectural Concepts” a separate document, which will detail the computing infrastructure.

tion of assets, asset service registries, asset services and software applications. This also includes capabilities to manage the evolvement of asset service registries, asset services and software applications over the time properly. More details are mentioned in the corresponding activities, see section "Activities".

- The computing infrastructure has to guarantee secure access to the uniform asset services, following the policies defined in the various asset service registries, asset services and software applications by the various organizations having an interest in the assets.

As described in the section "Overall Roadmap", we can distinguish between different levels of exploration capabilities provided by a computing infrastructure. The core applications considered in this paper require the following exploration capabilities:

- Publication and exploration of assets: The organization having an interest in an asset knows his own assets, thus, there exist exploration capabilities for the own assets ("own asset registry"). In addition, some software engineer publishes in a "global asset registry", which organization may see and explore this asset. Furthermore, a software engineer can share asset relations of this asset and such relations can be explored by authorized software applications resp. asset services via the asset.

- Publication and exploration of asset services (which addresses the publication and exploration of asset-oriented or asset-related data): A software application resp. asset service can explore each asset registry belonging to their organization as well as the asset services of other organizations published for their organization.

- Publication and exploration of organizations: An organization knows all organizations, where it has a contractual relationship ("own organization registry"). In addition, an organization publishes to what extent it wants to be known in a "global organization registry". Of course, the operator of the computing infrastructure must know all involved organizations, but the computing infrastructure could consist of multiple computing infrastructures that are technically decoupled (e.g. MindSphere ecosystem versus SAP ecosystem).

On this basis, it is then possible to further expand the exploration capabilities to be able to realize future brokerage services. There is the opportunity to design various brokerage roles including subscription to a "global asset registry" and "global organization registry". Nevertheless, each organization explicitly reveals what it wants to make available to a broker.

Parties

A party is an agent executing tasks by assuming a role. Parties strongly depend on the concrete application context of the asset administration shell. Therefore, we do not address the association of parties in this paper.

Activities

In this section, we describe some core activities. We break down the various activities in different clusters. For the description we use the concept as proposed by the Industrial Internet Consortium, see Figure 6. As a prerequisite we assume that there is available a computing infrastructure providing a ready-to-use initial setup.

Design and Integration of Asset Administration Shells

Activity "Scoping and modeling"

Triggers: The scoping and modeling of an asset administration shell are explicitly initiated by the business manager of the software engineer. This especially includes defining the *purpose* for the asset administration shell.

Workflow

- Task 1 "Definition of the asset service registry (declaration of associated asset services and relevant relations to other assets)": role software engineer

- Task 2 "Definition resp. modeling of the asset services": role software engineer

- Task 3 "Definition of initial setup": role software engineer

- Task 4 “Definition of access rights managed by the asset service directory (access rights of software applications to asset services, access rights to relations of this asset to other assets)”: role software engineer

- Task 5 “Definition of access and usage policies of the asset services”: role software engineer

Comments

- It is possible that an asset supplier delivers significant definitions and specifications directly with the asset, but independently of this, the software engineer has to define the scope and models relevant for his organization.

- Because of the online capabilities of the computing infrastructure with respect to creation, modification, and deletion of asset service registries and asset services we do not distinguish between creating new and modifying existing asset administration shells.

- Typically, in Task 1 the “header” of an asset service is declared (e.g. the name of the asset service and the necessary parameters, etc.). These declarations are listed in the asset service registry. In Task 2 the asset services are implemented, i.e. the bodies of the asset services are specified.

Activity “Implementation, deployment, and test”

Triggers: The initiation takes place explicitly by the software engineer after completing the scoping and modeling workflow.

Workflow

- Task 1 “Implementation of the asset services”: role software engineer

  - Task 1.1 Implementation of an asset service on the asset itself: in this case, typically the asset service is an encapsulation of capabilities delivered by the asset itself (e.g. measurement values and some business logic) and the asset itself has to provide appropriate programming capabilities. The computing infrastructure guarantees transparency with respect to the computing resources of the asset.

  - Task 1.2 Implementation of an asset service outside of the asset.

- Task 2 “Deployment of the asset services including following all requested policies, e.g. with respect to IT security”: role software engineer

  - Task 2.1 The asset service is deployed on the asset itself: in this case, the asset must first be integrated into the computing infrastructure (see Task 3.1 of activity “acquisition and commissioning of an asset: physical world asset”). The asset itself has to provide appropriate deployment capabilities.

  - Task 2.2 The asset service is not deployed on the asset itself.

- Task 3 “Testing the asset services in conjunction with the overall system consisting of software applications and asset service registries based on the computing infrastructure”: role software engineer

- Task 4 “Notification of all software applications and asset services with an interest in the new asset service registry and asset services”: role computing infrastructure

Effects: The asset service registry and asset services are ready to be used by software applications and other asset services.

Comments

- It is possible that an asset supplier directly delivers significant implementations of asset services. It may be that the software engineer only has to integrate the computing resources of the asset into the computing infrastructure; it may be that the software engineer only has to deploy the implementation, especially if the asset service will not be deployed on computing resources of the asset.

- Because of the online capabilities of the computing infrastructure with respect to creation, modification, deletion of asset service registries and asset services we do not distinguish between implementing, deploying and testing new and modifying existing asset administration shells.

Design and Integration of a Software Application

Activity "design, implementation, deployment, and test"

Triggers: This activity is explicitly initiated by the business manager of the software engineer.

Workflow

- Task 1 "Design of software application": role software engineer
- Task 2 "Implementation of the software application including a 'defensive' programming due to the possibility that access rights for asset services may change during operation": role software engineer
- Task 3 "Deployment of the software application including following all requested policies, e.g. with respect to IT security": role software engineer
- Task 4 "Test of the software application": role software engineer

Effects: The software application is ready to be used.

Comments

- Because of the online capabilities of the computing infrastructure with respect to creation, modification, deletion of software applications we do not distinguish between creating new and modifying existing software applications.

Usage of Assets

Activity "Acquisition and commissioning of an asset: physical world asset"

Triggers: This activity is explicitly initiated by the (intended) owner of the asset.

Workflow

- Task 1 "Acquisition of the asset": role owner of the asset
- Task 2 "Design of the asset administration shell of the asset": role software engineer belonging to the same organization as the owner of the asset (see corresponding activity)
- 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.
  - Task 3.2 The asset provides no own computing resources.
- Task 4 "Integration of the asset administration shell": role software engineer belonging to the same organization as the owner of the asset (see corresponding activity)

Comments

- Task 2: It is possible that an asset supplier delivers significant definitions and specifications directly with the asset, but independently of this, some software engineer belonging to the organization of the intended owner of the asset has to define the scope and models relevant for the organization.
- Task 4: It is possible that an asset supplier directly delivers significant implementations of asset services. May be that the software engineer only has to integrate the computing resources of the asset into the computing infrastructure ("plug & produce"); may be that the software engineer only has to deploy the implementation (especially if the asset service is not be deployed on computing resources of the asset).
- A software application or asset service will be notified about a new asset only, if the software application or asset service have internally modeled the possibility of new assets based on appropriate relations, for example a type-instance or implements-role relation. The notification itself is part of task 4 (resp. task 4 of the corresponding activity). If a human wants to be notified (for example in the case that a decision by a human is necessary), the software engineer has to create an appropriate software application.

- If the supplier of the assets has used the computing resources of the asset for the deployment of an own asset administration shell, then the new owner has to grant appropriate access and usage policies to the supplier.

Activity "Acquisition and commissioning of an asset: virtual world asset"

Triggers: This activity is explicitly initiated by some software engineer using an appropriate software application or triggered by some event and automatically executed by a software application.

Workflow

- Task 1 "Creation of the asset": role software application
- Task 2 "Creation of the asset administration shell of the asset"
  - Task 2.1 The asset administration shell is automatically created by the software application based on rules implemented by the software application using the capabilities provided by the computing infrastructure (example: P&ID engineering tool, which uses the concept of asset service registry and asset services to internally manage the engineering objects): role software application
  - Task 2.2 The asset administration shell is manually created (example: core objects of a request for proposal are managed based on the concept of asset service registry and asset services): role software engineer (see corresponding activities design and integration of an asset administration shell)

Activity "Modification of an asset: physical world asset"

Triggers: This activity is explicitly initiated by the owner of the asset.

Workflow

- Task 1 "Modification and commissioning of the asset (modification and commissioning in the real world)": role owner of the asset
  - 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 1.2 The modification does not affect own computing resources of the asset.
- Task 2 "Redesign of the asset administration shell of the asset according to the intended modification": role software engineer belonging to the same organization as the owner of the asset (see corresponding activity design of an asset administration shell)
- Task 3 "Integration of the redesigned asset administration shell, especially suitably deploying all requested asset services": role software engineer belonging to the same organization as the owner of the asset (see corresponding activity)

Comments

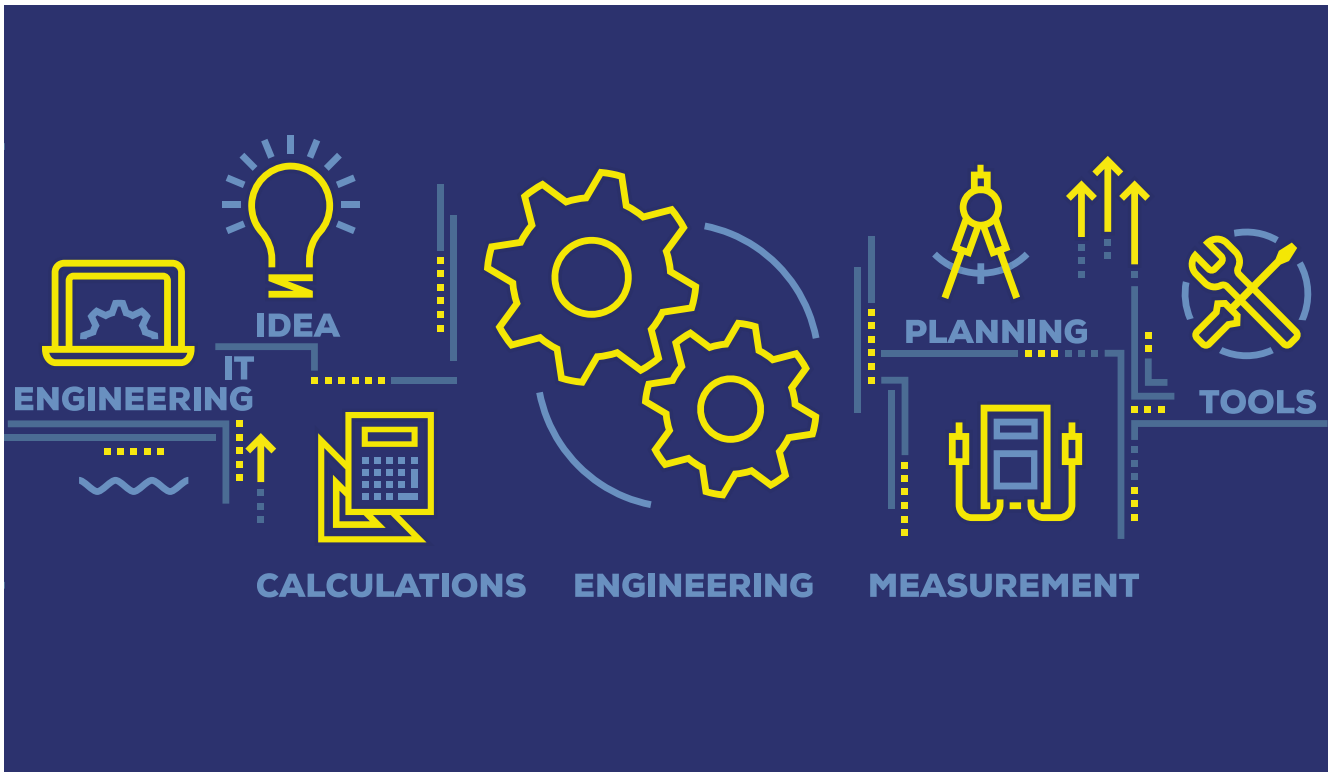
- Based on the notification subscriptions software applications and asset services belonging to other organizations will be automatically notified about the modification and can use these notifications to update values or trigger further actions.

Activity "Modification of an asset: virtual world asset"

Triggers: This activity is explicitly initiated by some software engineer using an appropriate software application or triggered by some event and automatically executed by a software application.

Workflow

- Task 1 "Modification of the asset": role software application
- Task 2 "Modification of the asset administration shell of the asset according to the intended modification: role software engineer (see corresponding activity design of an asset administration shell)"
  - Task 2.1 The asset administration shell is automatically modified by the software application based on



rules implemented by the software application using the capabilities provided by the computing infrastructure: role software application

- Task 2.2 The asset administration shell is manually modified: role software engineer (see corresponding activities design and integration of an asset administration shell)

Comments

- Based on the notification subscriptions software applications and asset services belonging to other organizations will be automatically notified about the modification and can use these notifications to update values or trigger further actions.

Standardization of Asset Services and Relations

Activity "Standardization of asset services and relations"

Triggers: Standardization activities are explicit initiated by a standardization organization.

Workflow

- Task 1 "Definition of asset services mandatory resp. optional for an asset (including implementation guidance for the asset services, commissioning guidance for assets and application policies for software applications) to be compliant with the standard": role standardization organization

- 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": role standardization organization

Effects: creation of a specific standard to be used in an application context

- If an asset $asset_n$ complies to a standard and is replaced by an asset $asset_m$ also complying to this standard (i.e. removing $asset_n$ and afterwards commissioning $asset_m$), then any software application is unaffected as far as it complies to the standard.

Comments

- There may be defined procedures to guarantee compliance of implementations with respect to a specific standard.

Provision and Operation of Computing Infrastructure

Activity "Provision and operation of computing infrastructure"

Triggers: This activity is explicitly initiated by the business manager of the operator of the computing infrastructure.

Workflow

- Task 1 "Development of computing infrastructure": role operator of the computing infrastructure
- Task 2 "Operation of computing infrastructure": role operator of the computing infrastructure

Effects: Provision of all capabilities requested to be provided by a computing infrastructure

Comments

- Depending on the application scenario the operator of the computing infrastructure may be restricted to a single company.

Miscellaneous Applications

Activity "Handling of intelligent assets with integrated asset administration shell"

Triggers: This activity is explicitly initiated by some software engineer.

Workflow

- Task 1 "Initial commissioning (identical to 'plug&produce' case in activity 'Acquisition and commissioning of an asset (physical world asset)')": role software engineer
  - When executing this task asset services can also be made available to other organizations.
- Task 2 "Removal from the computing infrastructure (see activity 'Modification of an asset (physical world asset)', whereby task 3 must ensure that all asset services, which should be furthermore available, are now deployed on a available computing resource of the computing infrastructure)": role software engineer
  - Example: The asset is temporarily in the warehouse for maintenance.
  - If necessary, information about the history of the asset has to be made available.
- 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 services are now back-deployed on the computing resource provided by the asset)": role software engineer
  - If necessary, actual information about the history of the asset has to be made available.

Boundary conditions

- The asset administration shell is provided with the asset, i.e. the asset service registry and asset services including a deployment on a computing resource provided by the asset.

Comments

- To standardize and automate this workflow, interaction protocols between asset administration shells can be used. This is currently being discussed in VDI/VDE-GMA Technical Committee 7.20 "Semantic and Interaction of I4.0 Components", see also [9], and will not be further elaborated here.

Activity "Adding and/or updating a specific parameter of an asset"

Triggers: This activity is explicitly initiated by some operation engineer, which might be an actual engineer at a factory line or an engineer for remote monitoring. The operation engineer will add and/or update such a specific parameter for an asset, for example updating the cyclic

period or defining a new event for data acquisition in a PLC (programmable logic controller), using some specific software application which is available, for example the configuration tool for the PLC.

Workflow

- Task 1 "Update of the asset administration shell of the asset according to the adding and/or updating of the specific parameter"
  - Task 1.1 The software application used for adding and/or updating the specific parameter takes care about the update of the asset administration shell of the asset: role software application
  - Task 1.2 The asset administration shell must be modified manually: role software engineer

Effects: Asset administration shell of the asset and parameter of the asset including their values are synchronized.

Comments

- The role software engineer in Task 1.2 can be executed by the operation engineer.
- There are many different possibilities how the manual modification of the asset administration shell in Task 1.2 is executed; this may range from the provision of some specific software application which provides appropriate capabilities to an operation engineer up to the necessity that a software engineer has to execute activities according to "design and integration of an asset administration shell".

Activity "Mass-updating of specific parameters in virtual world assets"

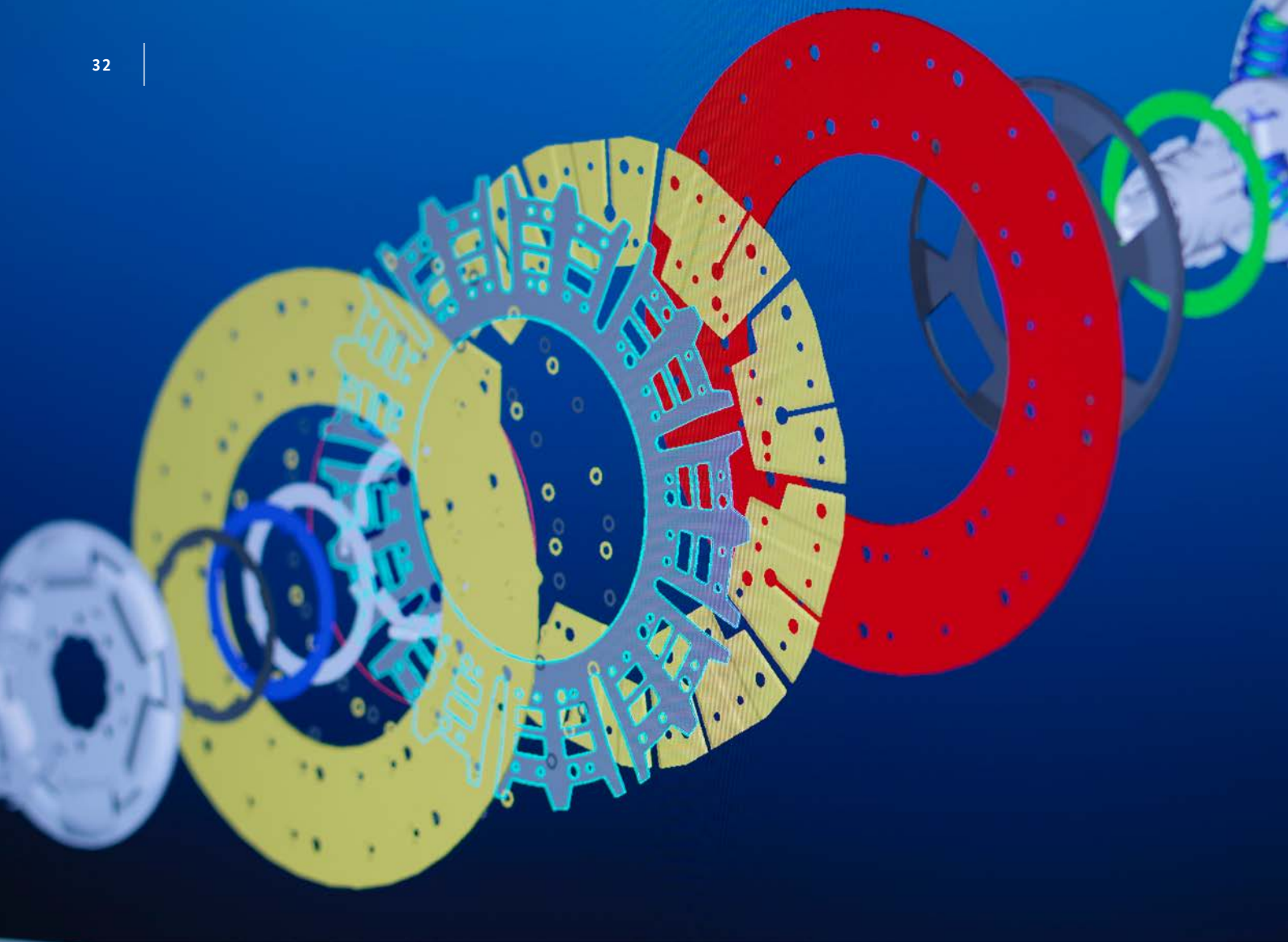
Triggers: This activity is explicitly initiated by some engineer, for example during engineering of a factory, where multiple number of same or comparable assets must be modified in a similar way (sometimes called bulk-engineering, i.e. instead of setting each parameter in each asset individually this is done automatically based on rules defined by an engineer). The engineer will do this using some specific software application, for example an engineering tool.

Workflow

- Task 1 "Define the target assets and define the specific parameter(s) and its appropriate value for updating": role user of software application
- Task 2 "Mass-update according to the definition": role software application (initiated by user of software application)

Comments

- We assume that the software application takes care about the update of the corresponding asset administration shells.



Relation to Application Scenario Value-based Service

For requirements engineering it is important to anchor the requirements – in this case both the activities and the various concepts of the system under consideration and the roles – in an explicit way to their source. Since there are many different application and usage scenarios for the asset administration shell, this document exemplifies one of them – specifically the application scenario Value-based Service. It is planned to consider further application scenarios also as soon as they are elaborated on a similar level of detail as the application scenario Value-based Service.

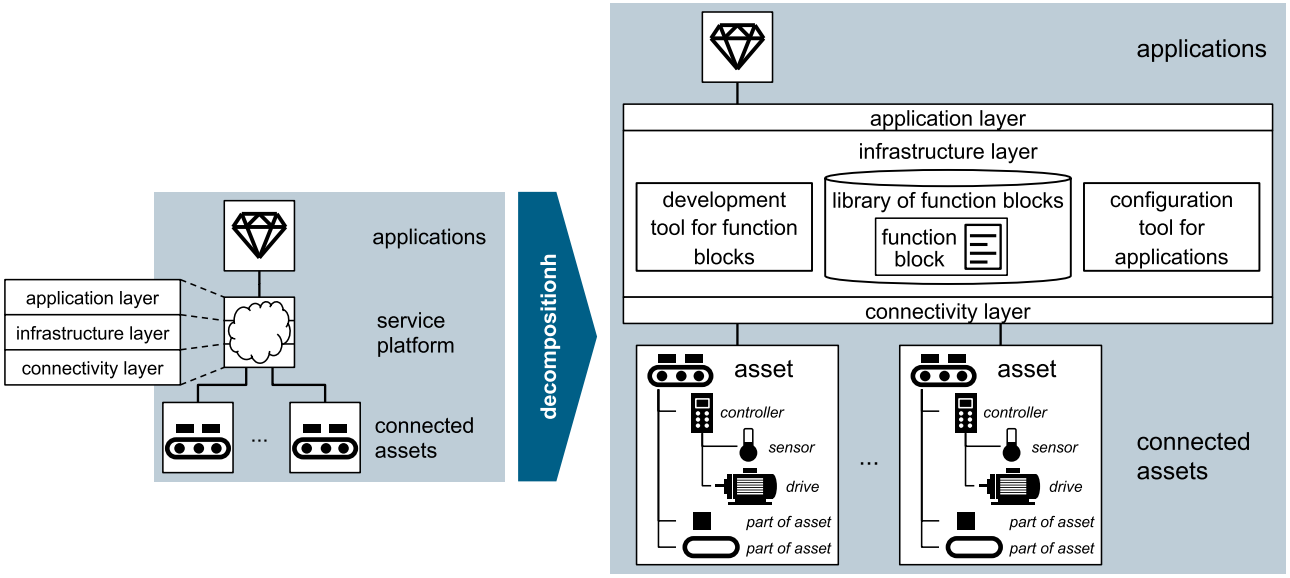
To link the usage view of the application scenario Value-based service according to [3] and [4], Figure 15 shows a more detailed representation of the system under consideration in the usage view Value-based Service. The individual connected assets are decomposed into their components and some of the essential components of the infrastructure layer are explicitly shown as described in [3].

To establish the link to the asset administration shell on this basis, one must adopt a complementary perspective in the form of a possible implementation of the system under consideration. This possible implementation follows an asset-oriented solution approach. This is shown in Figure 16.

In the context of such a possible implementation, the following relationships can be established:

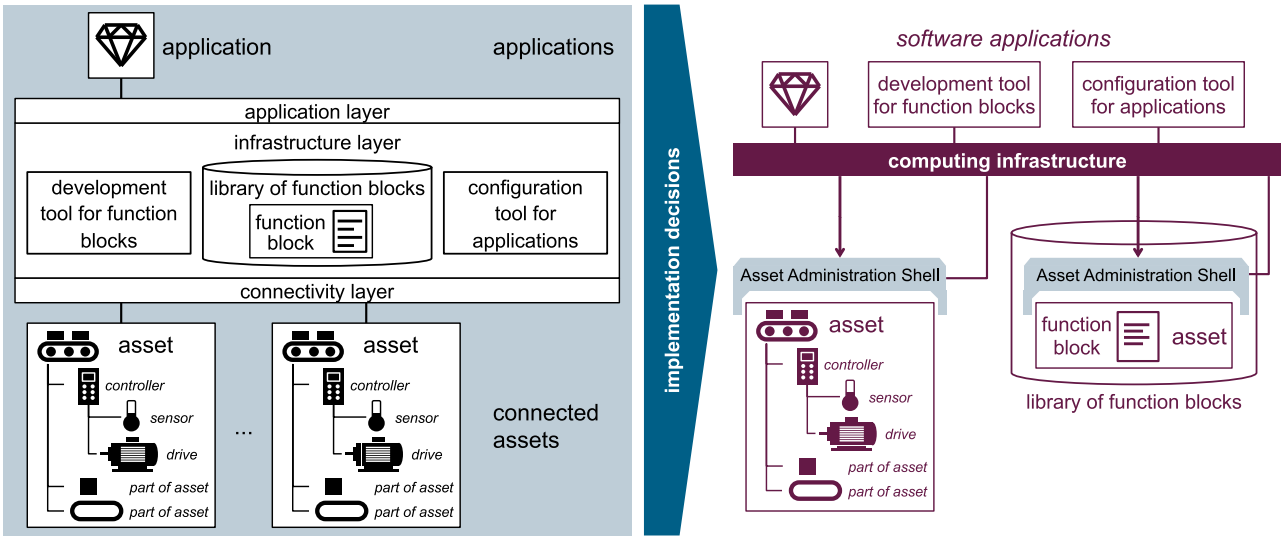
- The *asset* in the context of asset administration shell is a more generic concept of the asset according to the usage view of the application scenario Value-based Service.
- In addition to the connected assets according to the usage view of the application scenario Value-based Service, the function blocks can also be considered as assets of the virtual world with corresponding asset administration shells.

Figure 15: Detailing the system under consideration of usage view Value-based Service



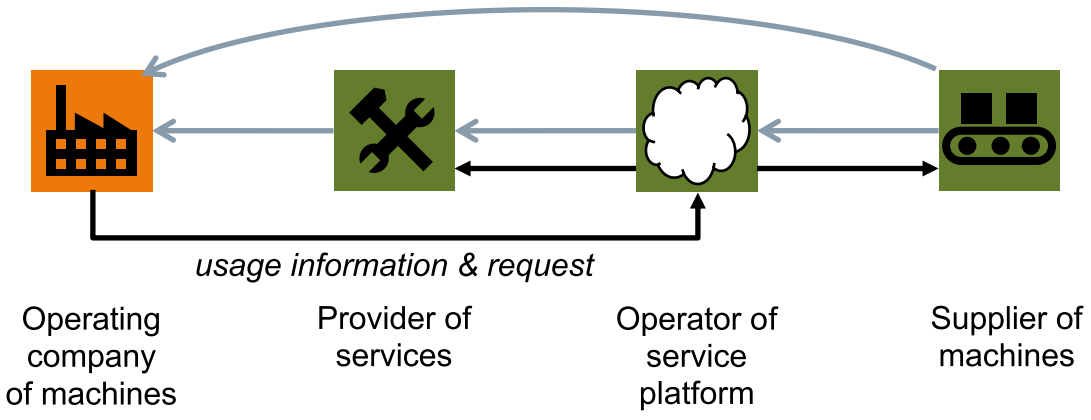
Source: Plattform Industrie 4.0

Figure 16: Implementation based on an asset-oriented solution approach



Source: Plattform Industrie 4.0

Figure 17: Considered business scenario of application scenario Value-based Services



Source: Plattform Industrie 4.0

- The library of function blocks could also be realized as an asset with its own asset administration shell[18].
- The applications according to the usage view Value-based Service are software applications in the sense of the usage view of the asset administration shell.
- The development tool for function blocks and the configuration tool for applications are software applications in the sense of the usage view of the asset administration shell. Of course it is possible to additionally model the development tool and the configuration tool as an asset with their own asset administration shells.
- Some function blocks could be structured according to assets and implemented on the basis of services following the principles of the usage view of the asset administration shell. Then these function blocks would implement specific asset services.

Based on specific activities, namely the activities "connection of an asset" and "development of a library of function blocks" according to [3], we will now illustrate the relation between the roles according to the usage view of the application scenarios Value-based Service according to [3] and the organizations having an interest in an asset and the roles "software engineer" as introduced in this document. As a business scenario, we base our illustration on the case that a supplier of a machine wants to offer its customers, i.e. the operating companies of the machines, additional services and operates an own service platform, see Figure 17 and also [2].

Figure 18 illustrates the activity "connection of an asset" according to [3].

In the usage view of the application scenario Value-based service there were introduced among others the following roles:

- Production manager: This is somebody from the organization "operating company of the machine", who defines the interests of this organization with respect to the connection of the machine, see "Triggers". In addition, the production manager finally is involved in the acceptance of the technical connection of the machine, see "Task 5".

18 This is not shown in Figure 16.

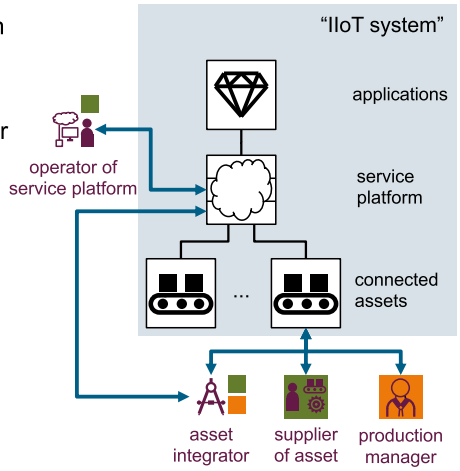
Figure 18: Activity "connection of an asset" according to [3]

Triggers: will be initiated and designed by the role production manager in an explicit way

Workflow

- **Task 1** "Define data to be transferred to service platform (including transfer protocol)": **role** asset integrator & supplier of asset
- **Task 2** "Connect asset to service platform": **role** asset integrator
- **Task 3** "Provide access to usage data of asset": **role** operator of service platform
- **Task 4** "Validate connection of asset": **role** asset integrator & operator of service platform & supplier of asset
- **Task 5** "Acceptance of connection of asset": **role** asset integrator & production manager

Effects: basic precondition for collecting usage-data of an asset



Source: Plattform Industrie 4.0

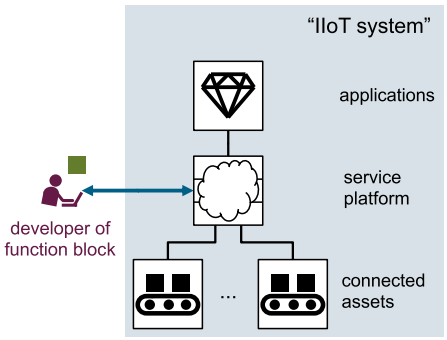
Figure 19: Activity "development of a library of function blocks" according to [3]

Triggers: will be requested by the business responsible of the developer of function block in an explicit way

Workflow

- **Task 1** "Design concept for library of function blocks": **role** developer of function block
- **Task 2** "Develop the function blocks": **role** developer of function block
- **Task 3** "Deploy and test the library of function blocks": **role** developer of function block
- **Task 4** "Continuous improvement of the library of function blocks": **role** developer of function block

Effects: basic precondition for configuration of applications



Source: Plattform Industrie 4.0

- Supplier of asset: This is somebody from the organization "supplier of the machine", who defines the interests of this organization with respect to the machine, see "Task 1".

- Asset integrator: This is somebody who implements the role software engineer according to this document. The asset integrator assumes both the role of a software engineer belonging to the organization "operating company of the machine" and the role of a software engineer belonging to the organization "supplier of the machine". Therefore, this role is labeled with both colors "orange" and "green".

- Operator of service platform: This is somebody who implements also the role software engineer according to this document and belongs to the organization "supplier of the machine", see Figure 17.

Figure 19 illustrates the activity "development of a library of function blocks" according to [3].

- The developer of function block is somebody who implements the role software engineer according to this document and belongs to the organization "supplier of the machine", see Figure 17. The business responsible of the developer defines the requirements which have to be implemented by the developer, see "Triggers".

In the description of the usage view of the application scenario Value-based Service, further technical activities as listed in Table 1, were explained. Table 1 gives an overview of the extent to which the concept of asset administration shell supports to implement these concrete activities:

Table 1 illustrates that the asset administration shell significantly supports the implementation of key ideas of the application scenario Value-based Service. This highlights the importance of the asset administration shell. However, it also illustrates that in the application scenario Value-based Service additional application-specific requirements must be considered, where the asset administration shell cannot contribute significantly.

Table 1: Technical activities described in the usage view of the application scenario Value-based Service

Technical activity	Impact	Rationale
Connection of an asset	●	Described in the activities "Usage of assets" (explained in the previous section)
Reconfiguration of an asset	●	Described in the activities "Usage of assets"
Configuration of an application	◑	Supported by the generic activity "Design and integration of a software application"
Development of a library of function blocks	◕	Strongly supported by the activities "Design and integration of asset administration shells" (explained in the previous section)
Development of a service platform	◔	Guidance provided by overall capabilities of the computing infrastructure
Collection and analysis of usage data of an asset	◕	Strongly supported by asset services, especially for assets providing own computing capabilities
Recording additional data on spontaneous request	◕	Overall workflow strongly supported by concept of asset administration shell
Operation and maintenance of a service platform	◔	Guidance provided by capabilities of the computing infrastructure
Generation of recommendations resp. requests for action	○	Application-specific
Execution of recommendation resp. request for action	○	Application specific
Benchmarking of machines	◔	Standardization of asset administration shell supports benchmarking

References

[1] Proposal for a joint “scenario” of Plattform Industrie 4.0 and IIC, https://www.plattform-i40.de/I40/Redaktion/DE/Downloads/Publikation/joint-scenario.html

[2] Benefits of Application Scenario Value-Based Service, https://www.plattform-i40.de/I40/Redaktion/DE/Downloads/Publikation/benefits-application-scenario.html

[3] Usage Viewpoint of Application Scenario Value-Based Service, https://www.plattform-i40.de/I40/Redaktion/DE/Downloads/Publikation/hm-2018-usage-viewpoint.html

[4] Functional Viewpoint of Application Scenario Value-Based Service, https://www.jmfrri.gr.jp/english/document/library/877.html

[5] German Standardization Roadmap Industrie 4.0, Version 3, https://www.din.de/blob/65354/57218767bd6da1927b181b9f2a0d5b39/roadmap-i4-0-e-data.pdf

[6] Aspects of the Research Roadmap in Application Scenarios, https://www.plattform-i40.de/I40/Redaktion/EN/Downloads/Publikation/aspects-of-the-research-roadmap.html

[7] The Industrial Internet Reference Architecture Technical Report, https://www.iiconsortium.org/IIRA.htm

[8] Glossary of Plattform Industrie 4.0, https://www.plattform-i40.de/I40/Navigation/DE/Service/Glossar/glossar.html & https://m.vdi.de/fileadmin/vdi_de/redakteur_dateien/gma_dateien/7153_PUB_GMA_-_Industrie_4.0_Begriffe-Terms_-_VDI-Statusreport_Internet.pdf

[9] Industrie 4.0-Sprache, https://www.plattform-i40.de/I40/Redaktion/DE/Downloads/Publikation/hm-2018-sprache.html

[10] Structure of the Administration Shell, https://www.plattform-i40.de/I40/Redaktion/EN/Downloads/Publikation/structure-of-the-administration-shell.html

[11] Relationships between I4.0 Components – Composite Components and Smart Production, https://www.plattform-i40.de/I40/Redaktion/EN/Downloads/Publikation/relationships-i40-components.html

[12] Industrie 4.0 Service Architecture: Basic concepts for interoperability”, https://www.vdi.de/fileadmin/vdi_de/redakteur_dateien/gma_dateien/END_3_Status_Report_Industrie_4_0_-Service_Architecture.pdf

[13] openAAS project, https://github.com/acplt/openAAS

[14] Industrie 4.0 Communication Guideline Based on OPC UA, https://www.plattform-i40.de/I40/Redaktion/EN/Downloads/Publikation/vdma-i40-communication-opc.html

[15] Guide for Writing Requirements, INCOSE 2017 (INCOSE-TP-2010-006-02.1)

[16] Details of the Asset Administration Shell Part 1 – The exchange of information between partners in the value chain of Industrie 4.0, https://www.plattform-i40.de/I40/Redaktion/EN/Downloads/Publikation/2018-details-of-the-asset-administration-shell.html

[17] Industrie 4.0 Plug-and-Produce for Adaptable Factories: Example Use Case Definition, Models, and Implementation, https://www.plattform-i40.de/I40/Redaktion/DE/Downloads/Publikation/Industrie-40-Plug-and-Produce.html

[18] Examples of the Asset Administration Shell for Industrie 4.0 Components – Basic Part, https://www.zvei.org/fileadmin/user_upload/Presse_und_Medien/Publikationen/2017/April/Asset_Administration_Shell/ZVEI_WP_Verwaltungschale_Englisch_Download_03.04.17.pdf

[19] Asset Administration Shell Concrete, in preparation

[20] The Structure of the Administration Shell: Trilateral Perspectives from France, Italy and Germany, https://www.plattform-i40.de/I40/Redaktion/DE/Downloads/Publikation/hm-2018-trilaterale-coop.html

[21] Reference model of Industrie 4.0 service architectures – Basic concepts and approach, https://www.degruyter.com/view/j/auto.2015.63.issue-10/auto-2015-0017/auto-2015-0017.xml

[22] DIN SPEC 16593-1: Reference Model for I4.0 Service Architectures Part 1: Basic Concepts of an Interaction-based Architecture, https://www.beuth.de/en/technical-rule/din-spec-16593-1/287632675

Annex: Relation to Other Activities and Publications

This chapter explains the relation of this document to various other publications in the context of the asset administration shell. Various terms such as asset service, asset service registry or computing infrastructure are introduced in this document, but they are *specific* in the context of this document. One should always keep in mind that these are *high-level concepts*. We emphasize that similar terms in a different context often have a completely different meaning.

General Remarks

According to the glossary of Plattform Industrie 4.0, see [8]

- an asset administration shell is a "virtual digital and active representation of an Industrie 4.0 component in the Industrie 4.0 system",
- an Industrie 4.0 system is a "system, consisting of Industrie 4.0 components and components implementation of a (standardized) communication", and
- an Industrie 4.0 component is a "globally uniquely identifiable participant with communication capability consisting of administration shell and asset within an Industrie 4.0 system which there offers services with defined characteristics".

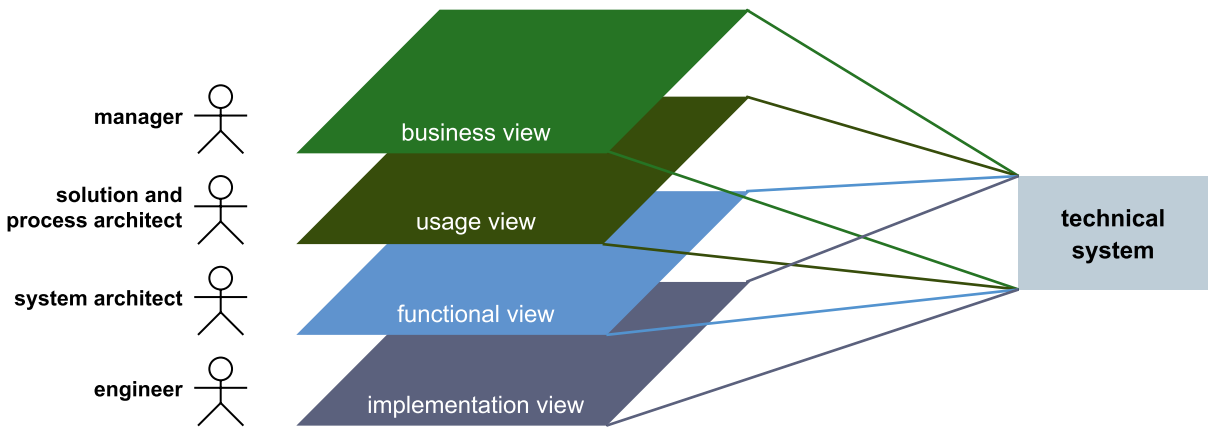
In this paper our focus of the asset administration shell is on its *description* and not on active representation in an Industrie 4.0 system. We assume that the so-called computing infrastructure offers capabilities to transform a description to an appropriate active representation of the asset administration shell.

The same applies to the term software application according to this paper, where our understanding also is a *description* and not an active representation.

If talking about usage, we mean a stakeholder who executes tasks related to such a description of an asset administration shell or software application. Typically, this will be a software engineer or software application engineer, but not a machine operator, production manager, or operator of a plant, for example.

We describe, on a conceptual level, how such software engineers handle these descriptions, how these descriptions are related to the associated asset, and how the computing infrastructure provides both the ability to transform these descriptions into appropriate active representatives and then execute them as intended.

Figure 20: Structuring principle, according to IIRA, see [7]



Source: Plattform Industrie 4.0

High level requirements according to this document are stakeholder and system needs according to INCOSE, see [15]. Typically, these needs are analyzed and formalized by stakeholder and system requirements, but this analysis is not in the scope of this document[19].

Overall Picture

To explain the relation to other documents, we use the Industrial Internet Reference Architecture, see [7]. It is based on the standard ISO / IEC / IEEE 42010 and suggests the structuring elements as shown in Figure 20.

Overall, some technical system is the object of consideration and is described from four different views:

- The business view identifies the business stakeholders and their business vision, values and objectives, or to put it in other words, describes the business models of the business stakeholders involved. Typically, managers are interested in and design this view.

- The usage view identifies stakeholders which interact with the technical system, or to put it in other words, describes a "black box" of the technical system in its usage context. Typically, solution and process architects are interested in and design this view.

- The functional view identifies the functional components, their interrelation and structure, the interfaces and interactions between them, and the relation and interactions of the technical system with external elements in the environment. Typically, system architects are interested in and design this view.

- The implementation view identifies technologies needed to implement functional components, their interaction and communication schemes and their lifecycle procedures. Typically, engineers are interested in and implement this view.

Typically, each view generates benefits to specific stakeholders, but usually the various views address *different* stakeholders. The content of a view is a *design decision*. Usually the individual views are not independent but *interlinked*.

The Industrial Internet Reference Architecture also proposes a description methodology for the various views in the form of so-called viewpoints. In this document we use the IIRA description methodology for the usage viewpoint; all other documents use their own description methodology.

We now apply this structuring principle to the asset administration shell, i.e. the object of consideration is the asset administration shell. Figure 21 shows an overview of some documents in the context of the asset administration shell following this structuring principle:

First of all it is important to mention that the different documents do not consider the same object of consideration. Some address the asset administration shell, whereas other documents address basic concepts to implement the asset administration shell.

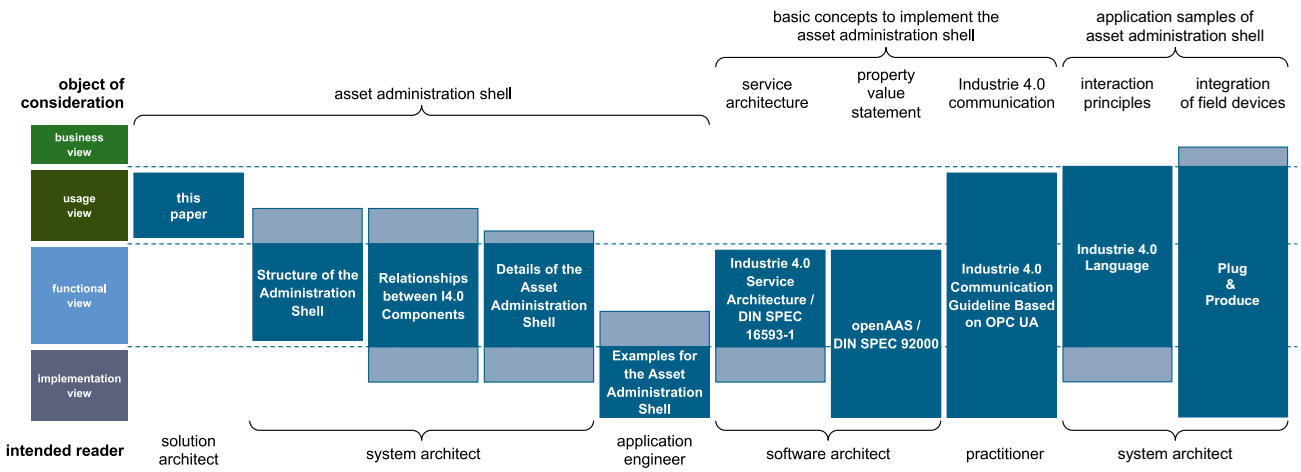
- There is no description of a business view of the asset administration shell and we do not intend to do this[20]. We think that the core of the asset administration shell is a *design* concept and that this concept will generate benefits in various business scenarios. It is important to understand these business scenarios to be able to link the usage view with a business rationale. Therefore, we explained this in chapter "Relation to Application Scenario Value-based Service" for one specific business scenario and we also intend to link the usage view of the asset administration shell to another application scenario later. In addition, the documents [9] and [17] also include relations to a business view.

- This document proposes a usage view of the asset administration shell from the perspective of a solution and process architect. It describes high-level concepts of systems and processes. In order to further detail these high-level concepts in the form of a functional view, one would especially have to detail the computing infrastructure.
  It is in the responsibility of a specific implementation activity to decide, which of these high-level concepts will be implemented and which not. Such implementation activities have to balance efforts and benefits; therefore, we do not claim that an implementation activity has to implement *all* high-level concepts described in this paper. We also do not claim that the high-level concepts are complete; we only claim that our description is

19 The requirements listed in the Annex of [16] are system resp. system element requirements according to [15].

20 Thus, business needs and business requirements according to [15] are not in the scope of this document.

Figure 21: Relation to other activities and publications



Source: Plattform Industrie 4.0

balanced in the sense that these concepts describe a comprehensive picture.
As an overall conclusion, it should be noted that we have identified no conceptual conflicts between this paper with the other documents.

- The other documents mainly focus on various functional views of the asset administration shell or even other objects of consideration, which will be explained in more detail in the following chapters.

As already mentioned, the different views typically address different stakeholders and therefore the documents address different readers. Regarding the document "Industry 4.0 Communication Guideline Based on OPC UA" it should be mentioned that especially this one addresses other readers compared to the other documents.

Structure of the Administration Shell

The document "Structure of the Administration Shell: Continuation of the Development of the Reference Model for the Industrie 4.0 Component", see [10], was published in April 2016. An update was published in March 2018 in the context of the trilateral cooperation between France, Italy and Germany, see [20], nevertheless we explain the relation based on [10].

It is based on the glossary of Plattform Industrie 4.0, where the asset administration shell is a virtual digital and active representation, thus, it mainly addresses functional aspects of the asset administration shell. There are described use cases, which are typical applications in the manufacturing industry addressing the typical acting people and manufacturing systems in a manufacturing company, but in these use cases the concept of asset administration shell is not mentioned in an explicit way. Thus, these use cases address aspects of the usage view. In addition, chapter 4 describes a methodology for the distributed formulation of sub-models, which also addresses the usage view.

Relationships between I4.0 Components

The document "Relationships between I4.0 Components – Composite Components and Smart Production: Continuation of the Development of the Reference Model for the I4.0 SG Models and Standards", see [11], was published in June 2017.

The focus of the document is on the internal structure of the asset administration shell; therefore, it mainly addresses functional aspects of the asset administration shell. The various relations between assets described in the document are *specific* relations between assets according to this document. Chapter 3.3 illustrates a use case and therefore addresses aspects of the usage view. At the end of chapter 4 also implementation aspects are addressed. Chapter 5 describes a modeling example addressing the functional view.

Details of the Asset Administration Shell

A first version of the document "Details of the Asset Administration Shell Part 1 – The exchange of information between partners in the value chain of Industrie 4.0", see [16] was published in December 2018. The document formally stipulates a few structural principles of the asset administration shell. Therefore, the core of the document addresses functional aspects of the asset administration shell. The target reader of the document is a development department in the value creation networks in order to have enough detailed information to start work on internal systems, whereas this document addresses a different target group.

A more detailed relation of the document "Details of the Asset Administration Shell" to this document is as follows:

- Chapter 2 introduces the leading picture. From a technical perspective the leading picture is more specific than the value chains considered in this document. In addition, the leading picture has a focus on design and engineering value creation processes, whereas the application scenario "Value-based Service", which is considered in this document, focuses on service value creation processes. The figure "File exchange between two value chain partners" illustrates an entity "user", which can act as such a "user". In this aspect, this document is more precise and distinguishes a software engineer and a software application. However, this should be obvious, as this document addresses a usage view. On the other hand, the figure "File exchange between two value chain partners" also shows a system boundary, whereas this document only considers a logical concept of a computing infrastructure.
- Chapter 3 introduces the meta model of the asset administration shell. The type-instance between assets is a specific kind of an asset relation according to this document. The description of identifiers addresses a specific aspect of a functional view. The concept "submodel-element" *details* the high-level concept of asset services. Collecting several sub-model-elements in a sub-model can be modeled by appropriate annotations of the corresponding asset services. Furthermore, the concept "views" can be modeled by appropriate annotations of the corresponding asset services. Finally, the concept "dictionary" is an aspect of detailing asset services in the context of activities of standardization organizations according to this document.
- Chapter 4 defines mappings to data formats and chapter 6 defines a package file format for the asset administration shell. Thus, these chapters address an implementation view.
- Chapter 5 describes access rights and a role concept, which details the access and usage policies as introduced in this document as well as selected capabilities of the computing infrastructure.

Examples for the Asset Administration Shell

The document "Examples of the Asset Administration Shell for Industrie 4.0 Components", see [18], was published in April 2017. These concrete examples address the implementation view. Besides concrete examples the document also describes various concepts, which can be found in various other documents also. Therefore, we do not incorporate these concepts in our discussions here, because we discussed these concepts in the context of other documents already.

Currently a VDI/VDE-Guideline is prepared by VDI/VDE-GMA Technical Committee 7.20 "Semantic and Interaction of Industrie 4.0-Components", see [19]. This guideline provides concrete examples for asset administration shells based on concrete implementation technologies. Therefore this guideline addresses the implementation view. In addition, some methodologies, which address the functional view, are described in the guideline.

Industrie 4.0 Service Architecture/ DIN SPEC 16593-1

In April 2014, the VDI/VDE-GMA Technical Committee 7.21 started to discuss aspects of service-orientation in Industrie 4.0 systems in a German VDI status report entitled "Industrie 4.0 – Auf dem Weg zu einem Referenzmodell", of which an English excerpt can be found in [21]. This paper provided an overview about the conceptual work performed by a working group of the VDI/VDE-GMA Technical Committee 7.21 towards the specification of a reference model for Industrie 4.0 service architectures. It focused on the conceptual design of service-oriented architecture (SOA) relying upon the "Industrie 4.0 component" as a core basic concept across the whole value network.

As a consequence, the document "Industrie 4.0 Service Architecture: Basic concepts for interoperability", see [12], was published in November 2016. In general, the document addresses a functional view related to the asset administration shell; the addressed target reader is a software architect. The term "service" refers to software services in a software service system. Therefore, in general the paper details the high-level concept "computing infrastructure".

Chapter 3 describes the relation to the asset administration shell and we want to illustrate the relation to the high-level concepts of this document: The asset administration shell is

- the sum of all information on an asset represented by the information models: this addresses a specialization of the high-level concept asset service
- which can be accessed through Industrie 4.0-compliant communication: this addresses capabilities of the high-level concept computing infrastructure
- and which can be understood through an Industrie 4.0-defined semantics or which follow a defined complementing data format: this addresses a specialization of the high-level concept asset service
- within a defined organizational scope: this addresses the high-level concept of belonging to an organization
- discoverable through a defined mechanism: this addresses the high-level concept asset service registry and capabilities of the high-level concept computing infrastructure
- based on common asset identification data: this addresses capabilities of the high-level concept computing infrastructure
- regardless of the deployment of the individual views on (other) assets in that domain: this addresses capabilities of the high-level concept computing infrastructure.

Chapter 4 describes the relationship to the Reference Model for Industrie 4.0 Service Architectures (RM-SA) document, a DIN SPEC workshop activity that finally resulted in April 2018 in the publication of the DIN SPEC 16593-1 document[21], see [22].

Chapters 5 – 7 describe the concretization (of aspects) of the computing infrastructure. Chapter 8 "Technology Mapping" addresses the implementation view.

openAAS project/DIN SPEC 92000

The openAAS project, see [13], was funded by ZVEI and developed core concepts necessary to implement the asset administration shell. In addition, the project provided a prototypical implementation of these concepts. After completing the project, the main concepts were published in DIN SPEC 92000.

The following concepts have been developed:

- A *property* describes intrinsic characteristics of an asset. Properties can be used to model static characteristics (e.g., technical data), dynamic online variables (e.g.,

21 The DIN SPEC 16593-1, see [22], describes basic concepts of an interaction-based architecture". It is part 1 of an intended series of DIN SPEC 16593-x finally leading to a "Reference Model of Industrie 4.0 Service Architecture (RM-SA)". It clarifies the use of the terms "service" and service-oriented architecture (SOA) in RAMI4.0. The DIN SPEC 16593-1 describes the classification into service types and the two major underlying interaction types: procedure-based interactions and state machine-based interactions to be used in Industrie 4.0 applications. By focusing on the interactions between Industrie 4.0 components, it provides a common conceptual foundation for both the service-oriented architectural style (see the *asset services* discussed in this document) and the message-oriented architectural style (see [9]).

measurements) or parameters (e.g., set points). There are provided specific services to manage *property value statements*, for details we refer to DIN SPEC 92000.

- A *lifecycle* entry is a time-stamped information entry associated to an asset, e.g. at time t_0 the parameter *setpoint* (i.e. a property of an asset) was set to *10*, at time t_1 the parameter *setpoint* was changed to *15*. Specific services are provided to create, delete, write, and read lifecycle entries.

Property value statements can be used to manage property values, unlike lifecycle entries to manage *arbitrary* entities, which may be defined in a less formal way.

The management of property value statements resp. lifecycle entries are *specific* asset services and therefore address specific functional aspects of the asset administration shell.

Industrie 4.0 Communication Guideline Based on OPC UA

The document "Industrie 4.0 Communication Guideline Based on OPC UA", see [14], was published in 2017 by VDMA.

This document addresses with Industrie 4.0 Communication a basic concept to implement an asset administration shell. In addition, it addresses with practitioner a different target group of readers than the other documents.

Industrie 4.0 communication as a concept addresses the functional view, the implementation based on OPC UA addresses the implementation view, and the described use cases as well as the proposed migration steps address the usage view.

Industrie 4.0 Language

The VDI/VDE-GMA Technical Committee 7.20 "Semantic and Interaction of Industrie 4.0-Components" is working on standardized interaction protocols between asset administration shells and defines a so-called Industrie 4.0 language, see [9].

The main idea is to design applications based on standardized interaction mechanism between asset administration shells. Thus, in this approach software-applications do not exist according to this paper.

The considerations are based on typical usage scenarios in manufacturing industries in the context of dynamic, self-organized, self-optimized and cross-company value networks. An example is the management of orders and generation of necessary manufacturing process steps, which is managed today typically in manufacturing execution systems. These considerations address the usage view.

The solution concept is based on declarative descriptions of vocabularies and interaction patterns, which are processed by generic inference mechanisms. This will typically result in less project specific integration efforts in engineering, commissioning and reconfiguration processes.

- Vocabularies are implemented by sub-models of asset administration shells (in this case a sub-model is a list of characteristics only).
- Interaction patterns are managed by a so-called interaction manager, which is a specific asset service according to this paper of each asset administration shell.
- In addition, there are asset services representing intelligent capabilities of an asset in the sense of being active and capable of making decisions. Such capabilities have to be classified appropriately, which is a task of a standardization organization according to this document.

The solution concept addresses the functional view. Finally, there are also some implementation considerations addressing the implementation view.

Plug-and-Produce for Adaptable Factories

The document "Industrie 4.0 Plug-and-Produce for Adaptable Factories: Example Use Case Definition, Models, and Implementation", see [17], was published in June 2017 by Plattform Industrie 4.0 in cooperation with ZVEI. The paper describes an application sample of the asset administration shell in the context of plug & produce of field devices.

We do not go into details and only mention the main relations in the context of Figure 21:

- Chapter 1 also contains an overall picture, but the scope of this picture is broader than the scope of Figure 21.
- Chapter 2 sketches the application scenario “Adaptable Factory” with focus on plug & produce of field devices and is therefore addressing aspects of a business view.
- Chapter 3 describes a usage view of plug & produce of field devices. The description is based on the template of IEC 62559-2 and is more detailed than a usage view description following the usage viewpoint of IIRA.
- Chapter 5 addresses a functional view of plug & produce of field devices and establishes the link to the asset administration shell. There is not only described the static structure of the asset administration shells involved, but also interaction sequences. In addition, the relation to service architectures according to [12] is explained and the subject of deployment of the asset administration shell is discussed.
- Chapter 6 and chapter 7 describe technology mappings and realization examples and therefore mainly address the implementation view.

AUTHORS AND GUESTS

From the German side the status report has been elaborated in the working group “Modeling Example” of the VDI/VDE-GMA Technical Committee 7.21 “Industrie 4.0 – Terms, Reference Models, and Architecture Concepts”, especially by the following members and guests:
Dr. Annerose Braune; TU Dresden | Prof. Dr. Christian Diedrich; Otto-von-Guericke-Universität Magdeburg | Dr. Sten Grüner; ABB AG, Ladenburg | Guido Hüttemann; RWTH Aachen University | Matthias Klein; Universität Stuttgart | Christoph Legat; Assystem Germany GmbH, München | Matthias Lieske; Hitachi Europe GmbH, Schwaig-Oberding | Dr. Ulrich Löwen; Siemens AG, Erlangen | Mario Thron; ifak – Institut für Automation und Kommunikation e.V., Magdeburg | Dr. Thomas Usländer; Fraunhofer IOSB, Karlsruhe

GUESTS

Alexander Belyaev; Otto-von-Guericke-Universität Magdeburg | Dr. Michael Okon; Fraunhofer IOSB, Karlsruhe | Prof. Dr. Gernot Spiegelberg; Siemens AG, Erlangen | Guido Stephan; Siemens AG, München | Jens Vialkowitsch; Robert Bosch GmbH, Feuerbach

CO-AUTHORS AND ADVISORS

From the Japanese side, some additional ideas were contributed by the following members of the “Use Case Task Force” in International Standardization Action Group, Robot Revolution & Industrial IoT Initiative:
Prof. Fumihiko Kimura; The University of Tokyo | Toru Ishikuma; Azbil Corporation | Dr. Hitoshi Komoto; National Institute of Advanced Industrial Science and Technology | Hirozumi Eki; JTEKT Corporation | Dr. Kiyotaka Takahashi; Hitachi, Ltd. | Takenori Baba; Mitsubishi Electric Corporation | Yutaka Manchu; Robot Revolution & Industrial IoT Initiative | Kazuo Nakashima; Robot Revolution & Industrial IoT Initiative

ADVISORS

Shinji Oda; Yokogawa Electric Corporation | Dr. Youichi Nonaka; Hitachi, Ltd.

ACKNOWLEDGEMENT

The authors would like to thank Dr. Heiko Koziolek (ABB AG) and Zhiman Chen (CRRC ZIC) for their valuable comments.

This discussion paper has been elaborated in the working group “Modeling Example” of the VDI/VDE-GMA Technical Committee 7.21 “Industrie 4.0 – Terms, Reference Models, and Architecture Concepts” together with the “Use Case Task Force” in the International Standardization Action Group, Robot Revolution & Industrial IoT Initiative. The activities and partners were initiated and orchestrated by the Standardization Council Industrie 4.0 (SCI4.0) within the project GoGlobal Industrie 4.0.

www.plattform-i40.de