

ICS 03.100.01; 25.040.01; 35.240.50

Ersatz für
DIN SPEC 16593:2017-04

**RM-SA –
Reference Model for Industrie 4.0 Service Architectures –
Part 1: Basic Concepts of an Interaction-based Architecture;
Text in English**

RM-SA –
Referenzmodell für Industrie 4.0 Servicearchitekturen –
Teil 1: Grundkonzepte einer interaktionsbasierten Architektur;
Text Englisch

RM-SA –
Modèle de référence des architectures de service pour l'Industrie 4.0 –
Partie 1: Concepts de base sur la base d'interaction;
Texte en anglais

Zur Erstellung einer DIN SPEC können verschiedene Verfahrensweisen herangezogen werden:
Das vorliegende Dokument wurde nach den Verfahrensregeln einer PAS erstellt.

Gesamtumfang 49 Seiten

Dieses Dokument wurde durch die im Vorwort genannten Verfasser erarbeitet und verabschiedet.

Contents

	Page
Foreword	4
Introduction.....	6
1 Scope	8
1.1 RAMI4.0 dimensions.....	8
1.2 SOA reference models	9
1.3 Value chains.....	10
2 Normative references	10
3 Terms and definitions.....	11
4 Symbols and abbreviated terms.....	14
5 Architectural foundation	15
6 Reference Models for Service-oriented Architecture.....	16
6.1 Overview	16
6.2 ISO/IEC 10746-1:1998 reference model for open distributed processing.....	16
6.3 OASIS Reference Model for Service-oriented Architectures.....	17
7 Conceptual interaction model.....	19
7.1 General	19
7.2 Basic considerations.....	20
7.2.1 Object worlds	20
7.2.2 Systems and components.....	20
7.2.3 Intra-component view: behavior, states, procedures and change events	22
7.2.4 Extra-component view: signal events, call events and interfaces	24
7.2.5 Inter-component view: primitives and notifications	25
7.2.6 Inter-component view: interactions, interaction patterns and interaction policies.....	25
7.3 Service.....	28
7.3.1 Definition.....	28
7.3.2 Service types.....	29
7.3.3 Infrastructure services	30
7.3.4 Interaction-based services.....	31
8 Characterization of interactions	36
8.1 Classification of interactions	36
8.1.1 General	36
8.1.2 Determinism.....	36
8.1.3 Synchronicity	36
8.1.4 State handling	36
8.2 Interaction patterns	37
8.2.1 General	37
8.2.2 Unicast.....	37
8.2.3 Multicast.....	38
8.2.4 Request-response	39
8.2.5 Multicast request-response	40
8.2.6 Use and observation	41
8.2.7 Publish-find-bind-execute.....	42
8.3 Characterization of interaction classes	43
9 Compliance.....	44

Annex A (informative) Terminological Service Discussion	45
Annex B (informative) Mapping examples to OPC UA	46
B.1 General	46
B.2 OPC UA read service	46
B.3 OPC UA PublishRequest	47
Bibliography.....	48

Figures

Figure 1 — RAMI4.0 dimensions and I4.0 component abstraction.....	8
Figure 2 — Reference models enable architecture specifications	9
Figure 3 — Influencing Factors of the RM-SA.....	15
Figure 4 — Concepts of OASIS SOA reference model	17
Figure 5 — IBA Conceptual interaction model	19
Figure 6 — RM-SA system model	20
Figure 7 — Component and its associated object worlds (derived from DIN SPEC 91345:2016-04, Figure 1) (left-hand side complete, right-hand side restricted for simplification reasons)	21
Figure 8 — Graphic representation of a UML state machine diagram (example)	23
Figure 9 — Projection of the functionality (in terms of procedures) and states of a component	24
Figure 10 — Exchange of primitives and notifications between components.....	25
Figure 11 — Interaction policies and domains.....	27
Figure 12 — Service types in an Industrie 4.0 System.....	29
Figure 13 — Mapping of interaction-based services to infrastructure services.....	30
Figure 14 — Infrastructure services according to ISO/IEC 7498-1:1994 OSI Model.....	31
Figure 15 — Spectrum of architectural styles of interaction-based services	32
Figure 16 — Functional services as remote operations.....	33
Figure 17 — Procedure-based interaction mapped to infrastructure services (note: illustration restricted to a procedure call).....	34
Figure 18 — State machine-based interaction mapped to infrastructure services.....	35
Figure 19 — Interaction pattern “unicast”	37
Figure 20 — Interaction pattern “multicast”	38
Figure 21 — Interaction pattern “request-response”	39
Figure 22 — Interaction pattern “multicast request-response”	40
Figure 23 — Interaction pattern “use and observation”	41
Figure 24 — Interaction pattern “publish – find – bind – execute”	42
Figure B.1 — OPC UA Read Service.....	46
Figure B.2 — OPC UA PublishRequest.....	47

Foreword

This DIN SPEC has been developed according to the PAS procedure. The development of a DIN SPEC according to the PAS procedure is carried out in workshops and does not require the participation of all stakeholders.

This document has been developed and adopted by the initiator(s) and authors named below:

- Fraunhofer IOSB
Dr. Thomas Usländer (initiator and workshop leader)
- Roland Heidel Kommunikationslösungen e.K.
Roland Heidel (deputy workshop leader)
- Assystem Germany GmbH
Christoph Legat
- Bosch Rexroth AG
Martin Hankel
- Fraunhofer IOSB-INA
Florian Pethig
- Fraunhofer IPA
Daniel Schel
- Hochschule Osnabrück
Prof. Dr. Clemens Westerkamp
- IBM Deutschland GmbH
Jörg Wende
- ifak
Prof. Dr. Christian Diedrich
- Karlsruhe Institute of Technology (KIT)
Julius Pfrommer
- MPDV Mikrolab GmbH
Ciske Busch
- PHOENIX CONTACT GmbH & Co. KG
Olaf Graeser, Dr. Lutz Jänicke
- Robert Bosch GmbH
Jens Vialkowitsch
- RWTH Aachen
Prof. Dr. Ulrich Epple, Torben Deppe
- Technische Universität Dresden
Prof. Dr. Martin Wollschläger

At present, there are no standards covering this topic.

DIN SPECs are not part of the body of German Standards.

A draft of this DIN SPEC has not been published.

Despite great efforts to ensure the accuracy, reliability and precision of technical and non-technical information, the Workshop cannot give any explicit or implicit assurance or warranty in respect of the accuracy of the document. Users of this document are hereby made aware that the Workshop cannot be held liable for any damage or loss. The application of this DIN SPEC does not release users from the responsibility for their own actions and is applied at their own risk.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. DIN shall not be held responsible for identifying any or all such patent rights.

Provision of this document free of charge as a PDF via the Beuth WebShop has been financed in advance.

Amendments

The main changes with respect to DIN SPEC 16593:2017-04 are listed below:

- a) Adoption of the “service” definition of OASIS RM SOA [2] whereby service behavior is described by means of interactions.
- b) Conceptualization of “service” with a spectrum of architectural styles instead of conceptualization of both “service” and “protocol”.
- c) Conceptualization of the Interaction-based Architecture (IBA) by means of UML 2.5 instead of informal illustrations.
- d) Use of UML 2.5 generic concepts (e.g. behavior, call event, signal event, message) wherever adequate.
- e) Service classification into the types “infrastructure service” and “interaction-based service” whereby
 - infrastructure services are described according to the layered ISO/OSI Reference Model and its definition of (N)-Services und (N)-Protocols, and
 - interaction-based services are described according to their behavioral description, being either procedure-based or state machine-based.
- f) Conceptualization of “interaction protocol” and “interaction pattern” as describing elements of “interaction policy”.
- g) Addition of the interaction pattern “publish – find – bind – execute”.
- h) Adaption of the terms and definitions to the new conceptualization.
- i) Addition of the terms “cooperation” and “collaboration” to the terms and definitions.

Previous editions

DIN SPEC 16593: 2017-04

Introduction

Standardization context

The exploitation of Industrie 4.0 (I4.0) paradigms in terms of economic benefit, flexibility and better-informed decision taking requires the availability of adequate information across all engineering and production value chains. Such information is the result of aggregation and fusion of functions applied to (big) data from various heterogeneous sources, often under real-time conditions and from production plants under constant change and different ownership. On the one hand, this comprises a challenge for the provision of an efficient, secure and dependable information management infrastructure. On the other hand, system architects must find solutions to assure interoperability on both syntactical and semantic level with reasonable engineering costs.

A common approach to solve this situation is the definition of a reference model that provides guidance to software and system engineers when defining system (of systems) architectures. The application of this approach for the service-oriented aspects of Industrie 4.0 is supported by the German Standardization Roadmap for Industrie 4.0 [1].

NOTE 1 This specification originates from the discussions within the Industrie 4.0 initiative, is motivated and guided by the Industrie 4.0 requirements and use cases and is aimed to be applied for the Industrie 4.0 working groups and standardization activities. However, it can also be applied beyond this initiative, e.g. in the context of the Industrial Internet of Things (IIoT).

A reference model encompassing the “big picture” of the system architecture is defined in DIN SPEC 91345, *Reference Architecture Model Industrie 4.0 (RAMI4.0)*. Service-orientation and service-oriented architecture (SOA) are mentioned several times in the description of the RAMI4.0 Layers (see NOTE 2 below) as one of the technological cornerstones of the I4.0 vision. Hence, this DIN specification describes the basic concepts of an interaction-based architecture (IBA) which concretizes what SOA and services mean in RAMI4.0.

NOTE 2 Throughout this document the RAMI4.0 Layers as defined in DIN SPEC 91345 are interpreted as a grouping of architectural aspects of an Industrie 4.0 system.

Motivation

According to DIN SPEC 91345 RAMI4.0, the “fundamental purpose of Industrie 4.0 is to facilitate cooperation and collaboration between technical objects” (synonymously called assets). RAMI4.0 considers assets of various types, ranging from field devices, control devices, work centers up to enterprises (following the classical automation pyramid, e.g. according to IEC 62264 [23] or ISA-95 [24]), but supplemented by the asset types product and connected world in order to enable new value chains. However, cooperation and collaboration activities are triggered and performed by I4.0 Components which constitute abstractions of individual assets. In order to enable the Industrie 4.0 vision of enhanced flexibility and adaptability, the hierarchical order scheme is relaxed on the abstraction level of the I4.0 Components. The ordering scheme has to be generalized and finally turns into a networked scheme of I4.0 Components.

This trend has consequences for the basic concepts of the resulting IT system architecture. In classical architectures following the automation pyramid, the information flows and roles of the communication participants are fixed to a large extent. The pre-defined architecture naturally leads to fixed communication schemes, e.g. following client/server or publisher/subscriber design patterns. Although these patterns can be combined to realize more complex scenarios, role changes and peer-to-peer schemes are difficult and inefficient to implement.

Having this in mind, this specification is guided by the following objectives:

- According to RAMI4.0 the architectural style to conceive and implement such a stack shall be service-oriented. Hence, in the light of the discussion above, there is a need to specify what is meant by service (oriented) architectures (SOA) to foster communication, interoperability, cooperation and collaboration between I4.0 Components.
- There is a need to consider if all interaction requirements in an I4.0 system, both for collaboration and cooperation, can be fulfilled by “services” or if other interaction forms will be required and used, too.
- There is a need to clarify the concept of “service” and relate it to basic concepts such as “interaction”, “interaction protocol” and “state transition”.
- There is a need to specify how services shall be specified and classified in classical service architectures.
- There is a need to define compliance to service and interaction protocol specifications.

The purpose of this document is to define basic concepts of an Industrie 4.0 system architecture such that these objectives can be fulfilled. Being Part 1 of DIN SPEC series 16593 (16593-1), this document proposes a so-called Interaction-based Architecture (IBA) aiming at considering the whole spectrum of “classical” SOA, message/rule-based protocols in order to respond to the diverse needs of Industrie 4.0 and new approaches [12]. Based upon this specification, further work can be started to provide service engineering guidelines for Industrie 4.0 architects and engineers.

It is envisaged to specify further parts of this DIN SPEC series following consolidated results of the acknowledged Industrie 4.0 working groups.

1 Scope

This document focuses on basic concepts of interactions between system components and therefore clarifies what the concept “service” stands for. As an initial document on this topic it facilitates the consensus finding within the Industrie 4.0 initiative.

The scope of this RM-SA DIN specification is given by the following three determinants:

- 1) the DIN SPEC of the Reference Architecture Model Industrie 4.0 (RAMI4.0) (subclause 1.1);
- 2) the purpose of a reference model dedicated to service-oriented architectures (SOA) (subclause 1.2);
- 3) the value chains to be considered in Industrie 4.0 (subclause 1.3).

1.1 RAMI4.0 dimensions

Looking at the RAMI4.0 and especially the three RAMI4.0 dimensions (see Figure 1), Industrie 4.0 services shall be applicable

- to all architectural aspects (RAMI4.0 Layers),
- to all hierarchy levels and, hence, all types of assets (from “product”, “field device”, “control device” up to the “connected world”), and
- to the whole lifecycle and value stream of both asset types and asset instances.

NOTE A “service” is considered as an asset itself, e.g. in order to enable service management in the same sense as the management of assets of other types. As a consequence, a service needs a specification of a manifest (service meta-data). This specification categorizes the structure and the mandatory information of such a manifest which enables service identification, discovery, orchestration and choreography.

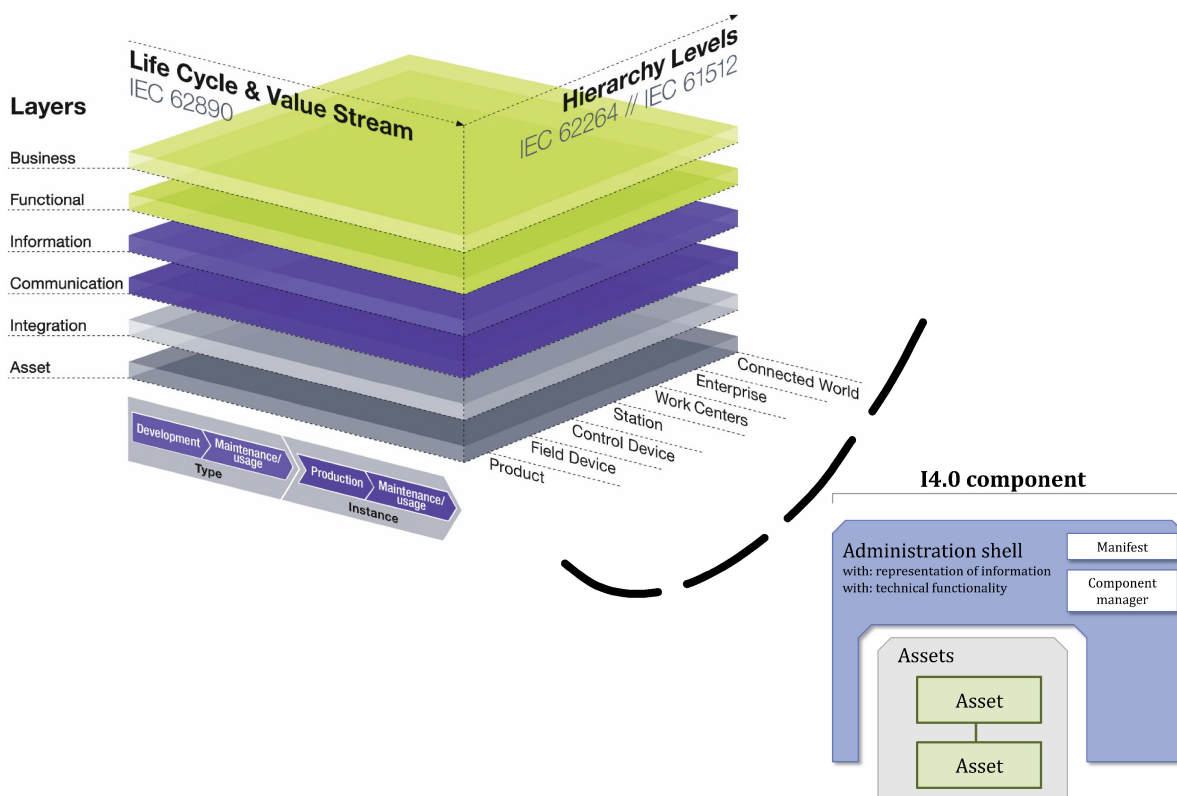


Figure 1 — RAMI4.0 dimensions and I4.0 component abstraction

1.2 SOA reference models

Following the terminology of the Organization for the Advancement of Structured Information Standards (OASIS), a reference model is an “abstract framework for understanding significant relationships among the entities of some environment. It enables the development of specific reference or concrete architectures using consistent standards or specifications supporting that environment.” [2]. Reference models set the conceptual foundation of distributed systems. Applied to service-oriented architectures (SOA), SOA reference models are meant to be meta-models for architectural specifications [3]. Hence, an RM-SA is an abstract framework that encompasses, among others, the

- identification and terminological definition of basic concepts (e.g. I4.0 Service) and their relationships within the scope of the Industrie 4.0 application domain,
- rules and notations to be applied,
- viewpoints to be described when defining Industrie 4.0 (reference) architectures, and
- statements that define what has to be fulfilled when claiming compliance to the RM-SA.

The RM-SA enables the specification of architectures, and this enablement encompasses both the conceptual and the technology-oriented implementation level. This is depicted in Figure 2.

In contrast to an architectural reference model, which defines the scope and the concepts that are basically to be considered in an architectural design, architectures derive from them the basic building blocks, their function, their relationships and their interfaces. This can be investigated just on conceptual level, i.e., abstracting from existing technologies, leading to conceptual architectures (left-hand side in Figure 2), or on technology level, leading to implementation architectures (right-hand side in Figure 2).

If such architectural specifications are open, based upon standards, solve commonly agreed requirements in a given domain (possibly derived from use cases, including their non-functional requirements), and can be implemented in an efficient and cost-effective way, they can be classified as reference architectures. Reference architectures “capture the essence of existing architectures and the vision of future needs and evolution to provide guidance to assist in developing new system architectures” [4].

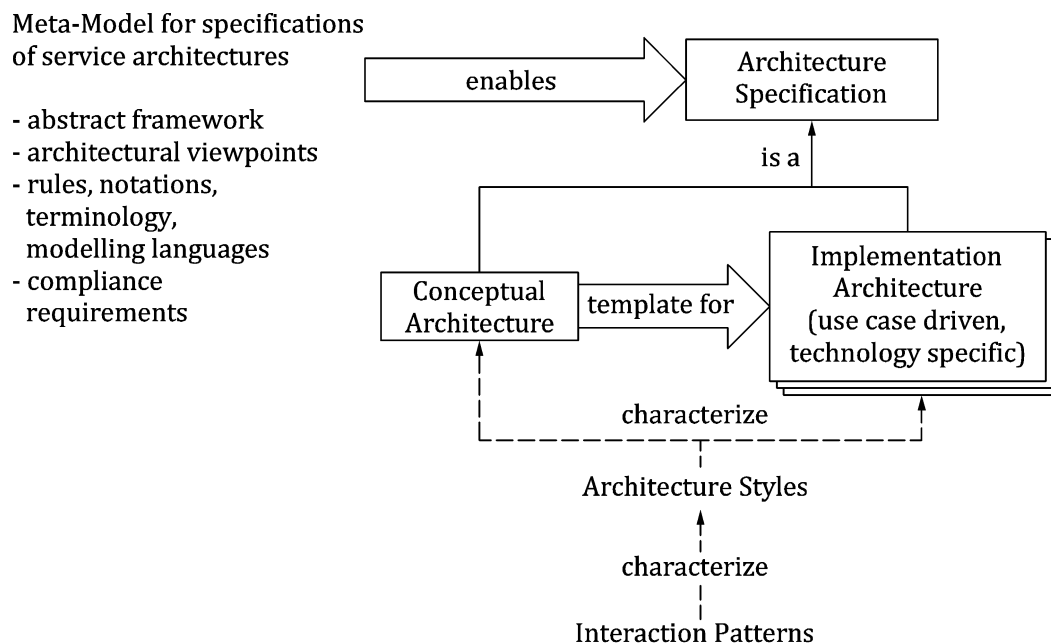


Figure 2 — Reference models enable architecture specifications

Both conceptual and implementation architectures are characterized by structural elements that are reusable in various instantiations and can act as blueprint specifications. In a SOA such structural elements are architectural styles.

The RM-SA defines an Interaction-based Architecture (IBA). As such it follows the idea of interactions as the basic conceptual elements that define service behavior (see below). Hence, the choice and combination of interaction policies and patterns determine the architectural style that is supported [5], e.g. message-oriented, event-driven, request/reply or resource-oriented (REpresentational State Transfer (REST) [6]). In 8.2, the RM-SA defines a list of basic interaction patterns that can be combined to define architectural styles. Its purpose is to streamline the discussions between the Industrie 4.0 working committees and support consensus finding.

1.3 Value chains

Looking from the application perspective, the scope of the RM-SA is defined by value networks comprising interconnected I4.0 abstract value chains [7], such as Product/Product Line Engineering, Process and Production Plant Engineering, Production and After Sales Services and Production Plant Construction and Operation.

Note that, having in mind that value chains will become much more flexible in the future, they do not fix any concrete value chain or exclude any other value chain options. Furthermore, value chains will be increasingly performed in a network of enterprises, which can lead to a change in the ownership of assets during the value chain. Access control mechanisms for information in the asset administration shell has to consider this.

Each value chain couples value processes along the lifetime of an asset. Note that there are additional accompanying value chains such as the edition and maintenance of standards as well as methods, tools and technologies, both in software and hardware. They are essential for the overall performance but not described here.

The role of the end-user who shall be able to influence product design, engineering and production in various processes of the value chains shall be emphasized, too. This will result in dedicated end-user services, whose needs have to be considered, too, when discussing the basic concepts of services in Industrie 4.0.

2 Normative references

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

DIN SPEC 91345:2016-04, *Reference Architecture Model Industrie 4.0 (RAMI4.0)*

ISO/IEC 7498-1:1994, *Information technology — Open Systems Interconnection — Basic Reference Model: The Basic Model*

ISO/IEC 10746-1:1998, *Information technology — Open Distributed Processing — Reference Model: Overview*

ISO/IEC 10746-2:2009, *Information technology — Open Distributed Processing — Reference Model: Foundations*

3 Terms and definitions

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

3.1

architectural style

coordinated set of architectural constraints that restricts the roles/characteristics of architectural elements and the allowed relationships among those elements within an architecture that conforms to that style

3.2

behavior

specification of events that can occur dynamically over time

[SOURCE: UML 2.5 [11]]

3.3

collaboration

type of composition whose elements interact in a non-directed fashion, each according to their own plans and purposes without a predefined pattern of behavior

[SOURCE: ISO/IEC 18384-1:2016 [20]]

3.4

cooperation

type of composition whose elements work or act together in a coordinated fashion on distinct tasks for a common purpose or benefit

3.5

contract

agreement by two or more parties

[SOURCE: OASIS RM SOA [2]]

3.6

event

set of possible occurrences. An occurrence is something that happens that has some consequence with regard to the system

[SOURCE: UML 2.5 [11]]

Note 1 to entry: This definition applies the event definition given in ISO/IEC 27000:2016 [21] to the IBA approach of this DIN SPEC. In ISO/IEC 27000 an event is defined as “occurrence or change of a particular set of circumstances” with the following notes:

- An event can be one or more occurrences, and can have several causes.
- An event can consist of something not happening.
- An event can sometimes be referred to as an “incident” or “accident”.

Note 2 to entry: It is also compliant with the event definition given in [19] which defines an event as “any happening of interest that can be observed from within a computer system”.

3.7

interaction

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

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

3.8

interaction domain

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

3.9

interaction pattern

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

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

3.10

interaction policy

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

3.11

interaction protocol

flow rules and the constraints in the exchange of messages (primitives and notifications) including the specification of their syntax and semantics and the roles of the interacting components

3.12

Interface of a component

projection upon a component that comprises 1) call events through which the call of procedures can be triggered, and 2) signal events that can be asynchronously observed by other components

Note 1 to entry: The definition of call events and signal events is taken from UML 2.5 [11].

3.13

I4.0 service

service that is handled as Industrie 4.0 asset

Note 1 to entry: Services are considered I4.0 services if they apply the rules for an Industrie 4.0 asset. They are represented in the information world by their own I4.0 component. This means that they have their own administration shell according to DIN SPEC 91345.

3.14

I4.0 platform

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

3.15

message

communication from one behavior to another requesting an operation call or signal reception

[SOURCE: UML 2.5 [11]]

3.16

notification

data unit that reifies a signal event

Note 1 to entry: A notification represents the data about the signal event to be exchanged with other components.

3.17**occurrence**

something that happens that has some consequence with regard to the system

[SOURCE: UML 2.5 [11]]

3.18**operation**

specification of a behavior of a component in terms of procedures

3.19**policy**

represents some constraint or condition on the use, deployment or description of an owned entity as defined by any participant

[SOURCE: OASIS RM SOA [2]]

3.20**primitive**

data unit that reifies a call event request, result or error

Note 1 to entry: A primitive represents the data of a call event request, result or error to be exchanged with other components and finally leading to a procedure behavior. The resulting primitives are categorized as request primitive, result primitive and error primitive, respectively.

3.21**service**

mechanism to enable access to one or more capabilities where the access is provided by a prescribed interface and is exercised consistent with constraints and policies as specified by the service description.

[SOURCE: OASIS RM SOA [2]]

3.22**service-oriented architecture**

software architecture that follows a service-oriented architectural style

Note 1 to entry: The term “service architecture” is used as synonym.

3.23**service-oriented architectural style**

architectural style that restricts the roles, characteristics and allowed relationships of services and service consumers

4 Symbols and abbreviated terms

AAS	Asset administration shell
API	Application programming interface
I4.0	Industrie 4.0 or Industry 4.0
IBA	Interaction-based Architecture
IIC	Industrial Internet Consortium
IIoT	Industrial Internet of Things
IIRA	Industrial Internet Reference Architecture
IoT	Internet of Things
JMS	Java Message Service
MDA	Model-Driven Architecture
MMS	Manufacturing Message Specification
OASIS	Organization for the Advancement of Structured Information Standards
OGC	Open Geospatial Consortium
OMG	Object Management Group
OPC UA	OPC Unified Architecture
OSI	Open Systems Interconnection
PDU	Protocol data unit
QoS	Quality of Service
RA	Reference Architecture
RM	Reference Model
RAMI4.0	Reference Architecture Model Industrie 4.0
REST	REpresentational State Transfer
RM-ODP	ISO Reference Model for Open Distributed Processing
SDU	Service data unit
SOA	Service-oriented Architecture
SOA-RM	Service-oriented Architecture – Reference Model
SOAP	Simple Object Application Protocol ¹⁾
UML	Unified Modeling Language
WSDL	Web Service Description Language
W3C	World Wide Web Consortium

1) Usage of this acronym is deprecated.

5 Architectural foundation

According to the vision statement of the “Platform Industrie 4.0” the step towards I4.0 requires the “availability of all relevant information in real-time by the connection of all entities that are participating in a value chain” as well as “the capability to derive the optimal added value” at any process step in the chain. Entities can be humans, objects and systems. From the technological perspective, these ideas coincide with the application of the principles of distributed processing and the emerging Internet of Things and Services to the domain of industrial production. Hence, when aiming at the specification of an RM-SA, it is valuable to first check the applicability of the reference models of distributed processing and both Internet evolutions, respectively, and their compliance to the I4.0 requirements [8].

One of the key concepts of RAMI4.0 to be followed is the definition of an I4.0 Component as it provides the common abstraction means for all types of assets. Hence, it influences the architectural building blocks and finally the reference architectures for a component and service-oriented I4.0 infrastructure [8].

This version of the RM-SA specification focuses on the concept “service” as one of the major influencing factors of the RM-SA as illustrated in Figure 3 and currently understood in SOA initiatives. The other influencing factors will be tackled in future versions or parts of the RM-SA [8].

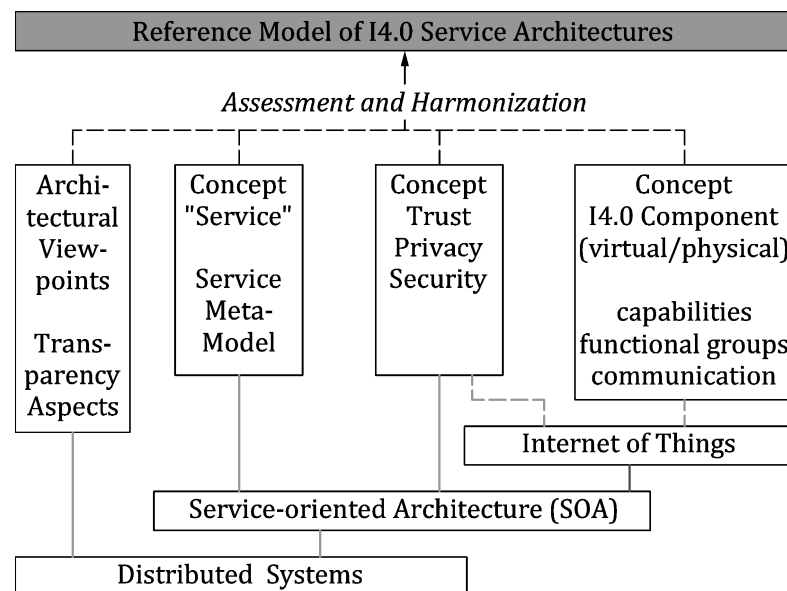


Figure 3 — Influencing Factors of the RM-SA