Methods

Michael Jacoby*, Friedrich Volz, Christian Weißenbacher, Ljiljana Stojanovic and Thomas Usländer

An approach for Industrie 4.0-compliant and data-sovereign Digital Twins

Ein Ansatz für Industrie 4.0-konforme und datensouveräne Digitale Zwillinge

Realization of the Industrie 4.0 Asset Administration Shell with a data-sovereignty extension
Realisierung der Industrie 4.0 Verwaltungsschale sowie einer Erweiterung für Datensouveränität

https://doi.org/10.1515/auto-2021-0074
Received May 14, 2021; accepted September 14, 2021

Abstract: Data sharing between enterprises requires both interoperability and data sovereignty. In the application domain of industrial production an integrated approach is required that encompasses standards and technologies of both Industrie 4.0 and the International Data Spaces (IDS). This paper describes how to combine them for the concept of Digital Twins following the architectural framework given in ISO DIS 23247. Furthermore, an implementation approach is described relying upon the Fraunhofer Advanced AAS Tools for Digital Twins (FA³ST). The resulting architectural approach may be combined with further open manufacturing standards, and may be applied for data analytics and the engineering of AI-based systems.

Keywords: Digital Twin, Industrie 4.0, data sovereignty, Asset Administration Shell, Industrial Data Space, FA³ST

Zusammenfassung: Datenaustausch zwischen Unternehmen setzt Interoperabilität und Datensouveränität voraus. In der Anwendungsdomäne der industriellen Produktion sollte daher ein integrierter Ansatz betrachtet werden, der Technologien sowohl der Industrie 4.0 als auch des International Data Spaces (IDS) umfasst. Dieser Artikel beschreibt, wie man diese Technologien nach dem in ISO DIS 23247 beschriebenen Architekturkonzept im Digitalen Zwilling vereint. Darüber hinaus wird ein Ansatz zur Implementierung, basierend auf den Fraunhofer Advanced AAS Tools for Digital Twins (FA³ST), beschrieben. Der daraus resultierende Architekturansatz kann mit weiteren offenen Produktionsstandard kombiniert und für die Datenanalyse und die Entwicklung von KI-basierten Systemen (KI-Engineering) eingesetzt werden.

Schlagwörter: Digitaler Zwilling, Industrie 4.0, Datensouveränität, Verwaltungsschale, Industrial Data Space, FA³ST

1 Introduction

The concept of a Digital Twin (DT) is a hot discussion topic in all initiatives that aim at conceiving and establishing service and data infrastructures for networked industrial production. Although not new, as the term and the concept has been already used more than twenty years ago in the context of product life-cycle management [1], it is getting raising attention due to the digitalization and, hence, virtualization of the physical assets in a production environment. In ISO DIS 23247-1 a DT is defined as "fit for purpose digital representation of an observable manufacturing element with a means to enable convergence between the element and its digital representation at an appropriate rate of synchronization" [2]. It should be distinguished between the concepts Digital Twin (DT), Digital Model (DM), and Digital Shadow (DS), that express different levels of integration between the physical and digital entities [1]:

***Corresponding author: Michael Jacoby,** Fraunhofer IOSB, Fraunhoferstr. 1, 76131 Karlsruhe, Germany, e-mail: michael.jacoby@iosb.fraunhofer.de, ORCID: https://orcid.org/0000-0002-1479-9242
Friedrich Volz, Christian Weißenbacher, Ljiljana Stojanovic, Thomas Usländer, Fraunhofer IOSB, Fraunhoferstr. 1, 76131 Karlsruhe, Germany, e-mails: friedrich.volz@iosb.fraunhofer.de, christian.weissenbacher@iosb.fraunhofer.de, ljiljana.stojanovic@iosb.fraunhofer.de, thomas.uslaender@iosb.fraunhofer.de

- a DM does not implement any form of automated data exchange between the physical and the digital entities,
- a DS implements an automated data stream between the state of the physical entity and the digital one, while,
- a DT implements automated data exchange in both directions between the physical and the digital assets.

Although we focus on DT in this paper, what is common to these concepts, is that an instance of such a digital representation alone is not helpful to support use cases of the production environment. There is a need to support interactions between DT instances, and to allow aggregation of DT instances to represent complex physical assets. Even more, as these DT instances (and their physical counterparts) may be installed in different organisational units, there is a need to consider supporting infrastructures for such networked DT instances. As a result, interoperability on syntactical and semantic level is not only a challenge for the data exchange between physical entities but also for the data exchange between digital entities. This also includes aspects of access control and usage control of data, hence, the question of data sovereignty [3].

In order to fulfil these requirements, the development of a DT-supporting infrastructure does exploit and integrate the concepts of the Platform Industrie 4.0, especially the Asset Administration Shell (AAS), and the International Data Spaces (IDS), especially the IDS Connector. This is the basic objective of this paper.

The remainder of this paper is structured as follows. Section 2 gives some background upon the AAS and the IDS, before Section 3 describes the approach for Industrie 4.0-compliant and data sovereign DTs following the architectural framework given in ISO DIS 23247 and based upon the Fraunhofer Advanced AAS Tools for Digital Twins (FA³ST). Section 4 presents a use-case before the paper is concluded with related work in Section 5 and a conclusion in Section 6.

2 Background

2.1 Asset Administration Shell

The Asset Administration Shell (AAS) is a concept developed by the Plattform Industrie 4.0, a network of companies, associations, trade unions, science and politics in Germany [4]. It is motivated by the Reference Architectural Model Industrie 4.0 (RAMI4.0) [5] and driven by the idea to manifest the concept of DT in factories and industrial production plants. The specification is currently published in two parts. Part 1 [6] introduces the AAS meta model including different serialization formats (JSON, XML, RDF, AutomationML, OPC UA node set, AASX) and Part 2 [7] defines different APIs in a protocol- and technology-agnostic way that can be used to interact with an AAS. As the specification is still evolving, these documents are expected to be updated in the future, e. g., an updated version of Part 2 specifying a mapping of the protocol-agnostic API to HTTP is expected in the near future. The AAS specification is subject to standardization in the document series IEC 63278 ED1 „Asset administration shell for industrial applications – Part 1: Administration shell structure“.

2.2 International data spaces

The International Data Spaces (IDS) is a data network focusing on data sovereignty, the ability of a data provider to determine who and how their own data can be used [8]. The central component, the IDS connector, is a gateway to the network and different implementations are in development to ensure data security and sovereignty. DIN SPEC 27070 specifies requirements and a reference architecture for security gateways, that ensure a trusted exchange of industrial data and services [9].

The application domains are broad and IDS communities focus on specific use-cases to support different industry branches. For example, the IDS-I (Industrial Community) aims to apply the IDS in the Industrie 4.0 (I4.0) [3]. For the technical enforcement of usage policies, usage control frameworks ensure correct application in all connectors. Our work is focused on the MYDATA [10] usage control framework in the IDS Trusted Connector [11]. The Trusted Connector is a reference connector implementation in the IDS with security and trust as the highest priority [8]. MYDATA is a proprietary implementation of a usage control framework which can be integrated into many IDS Connector implementations.

3 Approach

For use and integration of DTs across company borders, we propose to integrate DTs with the IDS to ensure security and confidentially of exchanged information. Figure 1 shows a conceptual integration of a DT with the IDS. From inside the company that owns/hosts the DT (i. e., Company A), an actor, which can be either human but typically

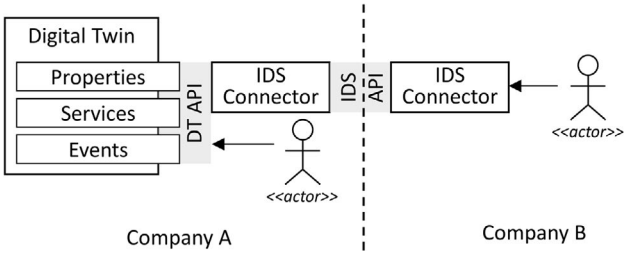

Figure 1: DT integration with IDS.

an application, can interact directly with a DT via the DT API. A DT exposes a well-defined DT API offering access to properties, services and events.

Actors from outside the company, i. e., from Company B, will have to use the IDS API to interact with the DT. This requires both Company A and B to each provide a (properly configured) IDS connector. Additionally, if the actor wanting to interact with the DT across company boundaries is an application, it needs to be certified according to the IDS Certification [12], similar to the connectors of both companies.

In the following, we describe our approach to realize I4.0-compliant and data-sovereign DTs. The proposed approach is based on the use of multiple specifications and standards such as the AAS specification for the DT meta model [6] and interfaces [7], ISO DIS 23247 [2] for defining the building blocks of each DT, and the IDS standard to ensure data sovereignty. This ensures an understanding of content and interfaces for communication with external systems within and across organizations, as well as for communication between components/models of a DT.

3.1 Realizing Digital Twins with FA³ST

FA³ST (Fraunhofer Advanced AAS Tools for Digital Twins) [13] is a Java-based software toolbox to create and manage AAS-compliant DTs. It comprises a library for developers to easily create and execute DTs, called FA³ST service, as well as the FA³ST registry which allows registering and discovering of DTs.

Figure 2 shows a high-level schematic diagram depicting the components and interfaces of the FA³ST service library. It is designed to be easily extendable and customizable by offering a variety of extension points. Core element is the *AAS Metamodel* which represents the DT with all of its properties and operations. By introducing the *Asset Connection* extension point on south-bound side, we support connecting the AAS to an asset which can be any kind of resource or legacy system independently

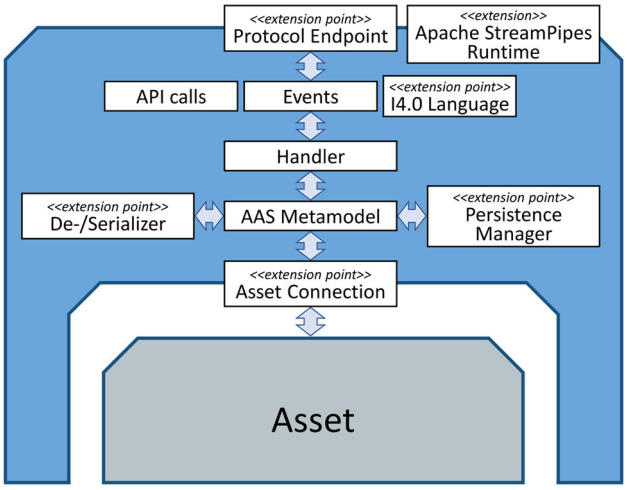

Figure 2: High-level schematic diagram of FA³ST service.

of the connection or protocol type. On northbound side, to realize an API to interact with the DT, we provide the *Protocol Endpoint* extension point, allowing to implement endpoints using multiple different communication protocols as envisioned by the AAS specification. FA³ST service provides a HTTP/REST-based and an OPC UA endpoint. Each endpoint can provide functionality defined by the different supported message types which may be *API calls* or *Events* (as defined in [6] and [7]) or may be formatted according to the *I4.0 Language* [14, 15] used by services with state machine-based interactions [16]. Mediation and synchronization between the endpoints and the metamodel happen in the *Handler* component. FA³ST service also supports multiple serialization formats via the *De-/Serializer* extension point as well as the use of different data storage systems via the *Persistence Manager* extension point.

As an additional feature, FA³ST service can be integrated with Apache StreamPipes [17] via the *Apache StreamPipes Runtime* extension. This enables adding stream processing capabilities to a DT allowing to compute additional properties of a DT at run-time. Apache StreamPipes provides an easy-to-use visual editor that supports non-expert users in creating so-called pipelines consisting of data sources, processors and data sinks. The data sources fetch data from external systems and feed it to the chain of processors. Data sinks can be used to send the result of the pipeline back to external systems. FA³ST already provides a DT source and sink component for Apache StreamPipes. More details on FA³ST can be found in [13] including a code example and details on the integration of FA³ST with Apache StreamPipes.

3.1.1 Alignment with Digital Twin reference architecture for manufacturing (ISO DIS 23247)

"The ISO 23247 series defines a framework to support the creation of Digital Twins" [18] in the context of the manufacturing domain. Part 2 of the series [18] defines a DT reference architecture for manufacturing on domain and entity level. It identifies the DT building blocks and provides guidance on how they should work together. This standard was used to design the FA³ST architecture as it ensures that all aspects are considered and each building block is well understood and correctly placed.

Figure 3 illustrates how the FA³ST service components introduced in Fig. 2 together with additional software components from the FA³ST ecosystem align with the entity-based DT reference model of ISO DIS 23247. The blue boxes represent the reference model entities from the ISO standard, the white boxes components of FA³ST, and dark grey boxes planned extensions to the FA³ST toolbox.

On the bottom of the figure are the observable manufacturing elements, i. e., the physical assets that should be represented by DTs. In the *Data Collection and Device Control Entity*, the *Asset Connection* of FA³ST spans across the *Data Collection Sub-Entity* and the *Device Control Sub-Entity* as it combines both functionalities.

According to the ISO standard, the *Core Entity* is composed of the *Operation and Management Sub-Entity*, the *Application and Service Sub-Entity*, and the *Resource Access and Interchange Sub-Entity*. The *Operation and Management Sub-Entity* "operates and manages" the DT and is also responsible for representation and synchronization [18]. This functionality is provided by the *AAS Metamodel*, *Persistence Manager*, *Handler*, and *De-/Serialization* in FA³ST. The *Resource Access and Interchange Sub-Entity* is responsible for communication of the DT with the outside world. In FA³ST, this comprises all *Protocol Endpoints* (see Fig. 2) represented by the concrete endpoints of HTTP and OPC UA protocol, as well as the extension point for future I4.0 languages and events.

The *Application and Service Sub-Entity* provides additional services that can be integrated with a DT, e. g., for simulation or analysis purposes. With FA³ST, we provide an integration with the Apache StreamPipes framework, via the *Apache StreamPipes Runtime* component.

"The User Entity can be any entity that utilizes the Digital Twin for manufacturing, including a human [...]" [18]. As part of the FA³ST ecosystem, we developed two components for the Apache StreamPipes visual editor, called DT Sink and DT Source, that enables non-expert users to use DTs as a data source and/or sink in their Apache StreamPipes pipeline with just a few clicks. Additionally, we are working on a web and a mobile application for users to interact with a DT.

The *Cross-System Entity* shown on the right is responsible for providing common functionality across the other entities, such as data assurance and security. FA³ST provides the *IDS Connector* component for these purposes.

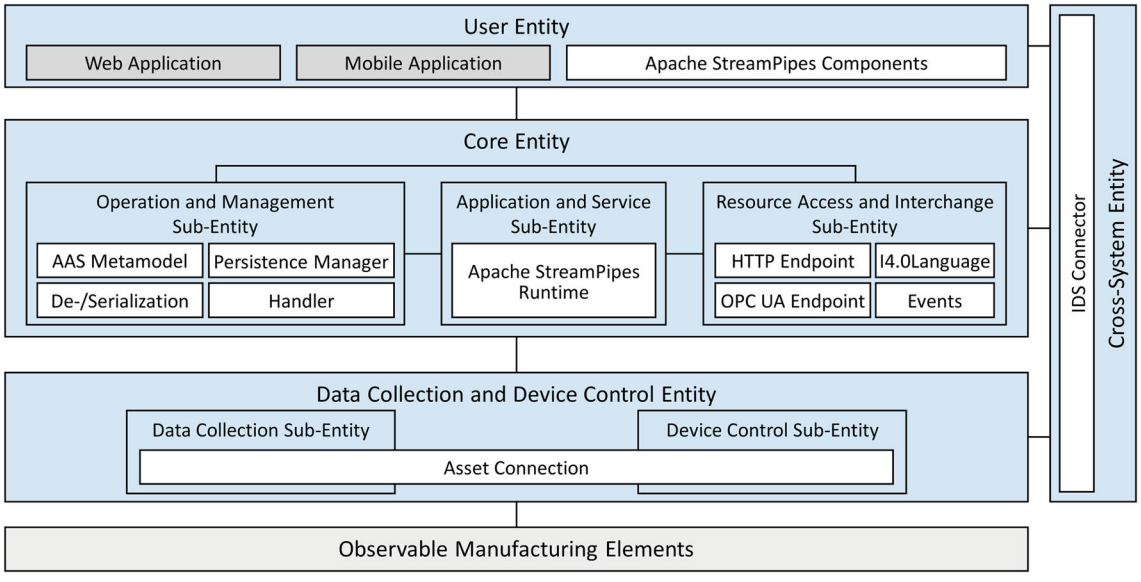

Figure 3: Alignment of FA³ST with the entity-based Digital Twin reference model for manufactoring from ISO/DIS 23247-2 [18].

3.2 Protecting Digital Twins with IDS

For the implementation of the I4.0-compliant DTs with FA³ST, the architecture of the IDS with their core component, the IDS Connector, is considered. In the context of ISO 23247-2, the IDS Connector is a *Cross-System Entity* used to realize data-sovereignty. Although the AAS specification basically proposes attribute-based access control (ABAC) for handling access control to data, there are only broad statements about the required service infrastructure to really support ABAC [6]. For example, the identity management solutions focus on the OpenID Connect framework. Data sovereignty includes usage control and data provenance tracking. Usage control extends the existing access control with additional obligations, that the data consumers need to follow. Provenance tracking describes the traceability of data in the network, allowing data owners to track their data even when it left their own connector [10]. Within the IDS-Industrial community, a joint activity between the Platform Industry 4.0 and the IDSA International Data Spaces Association (IDSA), data sovereignty requirements for the manufacturing industry are analyzed and mapped to joint architectural patterns that include a combination of AAS and IDS technologies [3].

However, the realization of a usage control system is a significant architectural endeavor and starting from scratch for the AAS would not be feasible for us. For example, the identity management is a central issue, which the IDSA already solved in its architecture [8]. The IDSA has several implementations of usage control frameworks in active development and demonstrated them in several use-cases [19]. These frameworks use Open Digital Rights Language (ODRL) policies which are then converted in implementation-specific policies [8]. While the specification of AAS policies currently focuses on access control, we defined ODRL policies for restricting usage of AASs and applied them with the IDS usage control framework "MYDATA Control" [10]. In future, the AAS policies defined with an AAS meta-model containing usage control rules have to be mapped to ODRL policies to be applicable in IDS usage control frameworks, in our case MYDATA. Another feasible approach in the future would be extending existing usage control frameworks or creating new frameworks to apply AAS usage policies directly in DTs. In this case a framework like MYDATA could be adapted to be used in I4.0-compliant DTs.

One fundamental key of the IDS architecture is that data resides inside of the IDS connectors, where usage control frameworks can control the data usage [8]. If data leaves the connector, it is difficult to enforce the correct usage. The IDSA is considering usage control outside of connectors by applying usage policies in the systems processing the data. In theory, AAS implementations like FA³ST could include its own usage control framework applying usage control rules. However, the IDS connector is designed as security gateway according to DIN SPEC 27070 and includes several security mechanisms like remote attestation and trusted platform support [8]. Current AAS implementations lack such features but will be enhanced as the implementations of DTs grow. In the current state, data is most secure inside the connector but looking at industry use-cases, it is unrealistic to expect companies to run all their software inside IDS connector containers including all of their I4.0-compliant DTs.

That is why we decided to keep most of the DT service outside of the IDS. Figure 4 shows the combination of FA³ST and IDS in alignment to ISO 23247. For clarity, the services inside of *User Entity* and *Core Entity* are not shown. The *IDS Connector* is part of the *Cross-System En-*

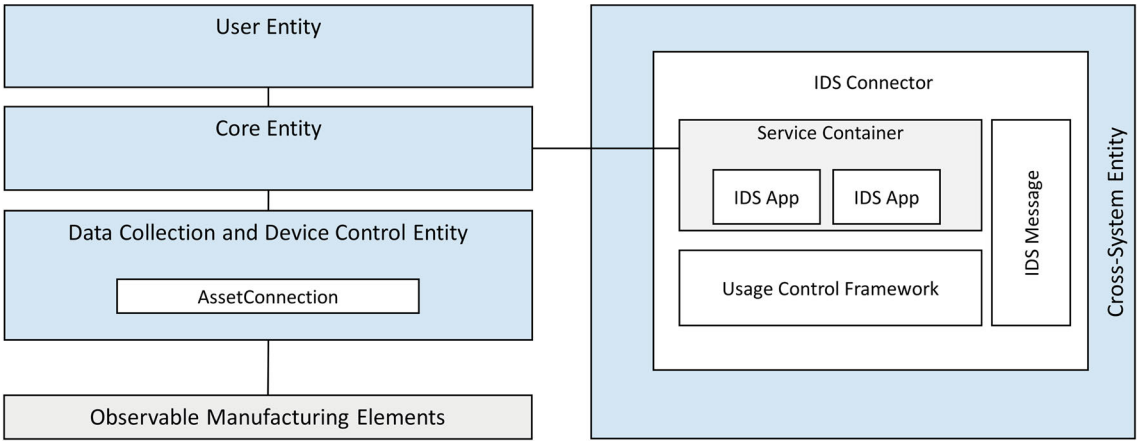

Figure 4: FA³ST in combination with the IDS aligned to ISO/DIS 23247-2 [18].

tity and contains a *Service Container*, *Usage Control Framework* and a component for the *IDS Message*. The *Service Container* contains isolated *IDS Apps*. The *IDS Message* component encodes and decodes messages for communication with other IDS connectors. Currently, the *IDS Messages* are multipart messages according to RFC7578 [8] but drafts for a generic IDS API [20] exist. DT services providing data are not part of the service container in the IDS Connector. DT services looking to process critical data, have to be deployed in the Service Container, but we usually extract the data processing part inside an IDS App to process highly critical data with usage policies, such as a business analytics app processing critical production parameter. In this case, the usage policy would ensure that the critical values are not leaked and only calculated KPIs are forwarded to other DT services.

In conclusion, we consider only running a subset of software processing highly critical data inside of IDS connectors as IDS apps. This can include IDS apps connecting to the core entity of the DT.

4 Use case "TableSort"

The FA³ST service and registry, which were introduced in Section 3.1, are applied in a demo scenario, which considers the use case of a coffee provider with an automatic sorting machine, TableSort [21], at its disposal. The sorting device is capable of separating desired from undesired coffee beans. In order to differentiate the beans, the system is measuring the properties of each coffee bean using a camera system such as size and degree of roast.

A potential customer should be able to place an order through a user interface, specifying the desired amount, roast and quality of coffee beans. This allows for automatically processing of incoming orders. After placing the order, the customer is kept informed about the progress. The provided information includes details such as the total progress, number of rejected beans as well as the order details.

In the considered scenario, as shown in Fig. 5, the sorting machine is represented by an AAS-based DT, which provides the AAS API specified in [7]. The AAS and AAS registry are implemented by the FA³ST service and registry. Connections to applications in external system environments are encapsulated through an IDS connector to ensure security as well as usage control of the outgoing data. The usage control policies include restrictions so that data will only flow into apps with certain purposes during certain time intervals. The asset connection provides the interface to the physical device, which realizes the data collection and device control entity described in [18]. In de-

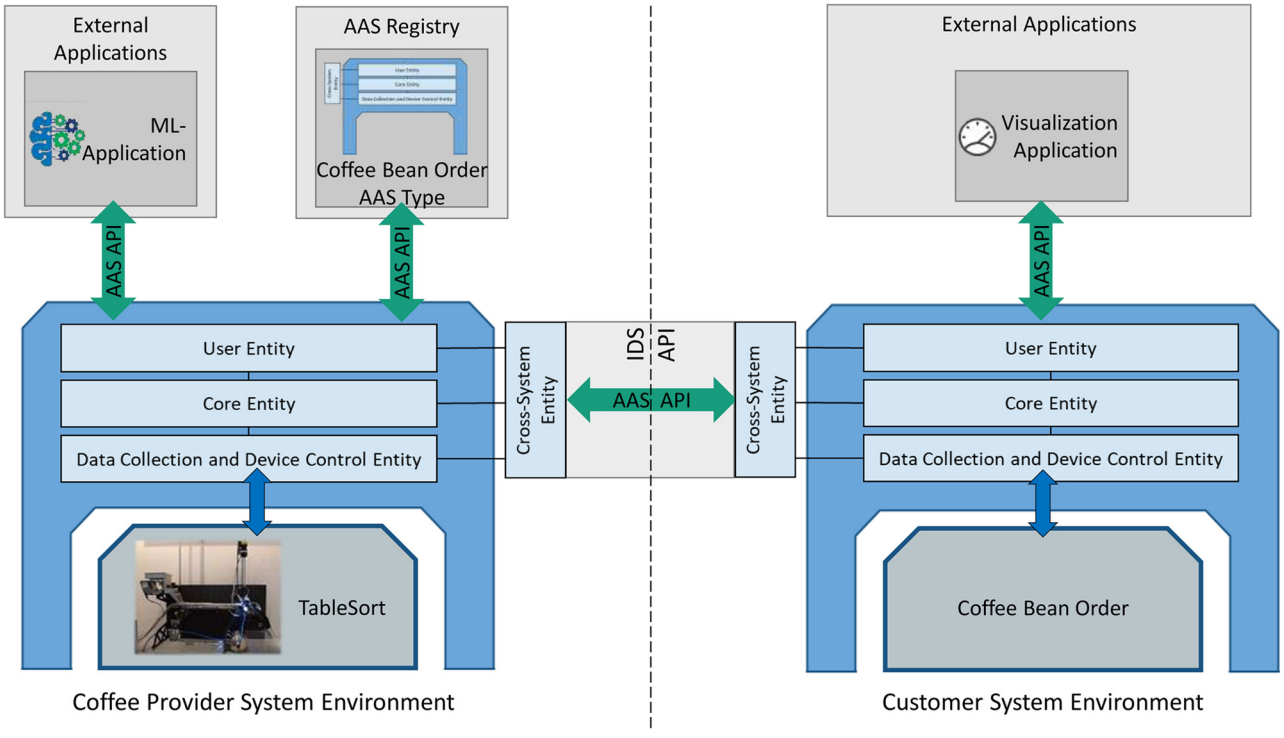

Figure 5: High-level architecture of "TableSort" use case.

tail, the asset connection implements the means to read and write, as well as initiating operations on the sorting device via an OPC UA server, which was extended to be I4.0 compliant [22].

If a customer places an order, the request is handled by the corresponding endpoint in the core entity, as the web and mobile applications in the user entity layer are not implemented yet (see 3.1.1). In the present scenario, in response to the placement of an order, the request is forwarded to the AAS registry. The AAS registry then provides the customer with an instance of the AAS type "coffee bean order", as an AAS is also capable of representing virtual objects such as an order. The instance is automatically created after the order has been placed, and deployed in the customer system environment. This instantiated AAS contains the specific order details and provides the necessary data for the external visualization application by connecting to the sorting machine. Figure 5 depicts the demo setup, with an already deployed AAS in the system environment of the customer. The asset connection allows to cache the information about the order progress within the TableSort AAS. This data can be accessed by the coffee bean order AAS using the IDS API, thus the data can only be processed following the terms specified in the data source usage policy. The number of AAS internally (within its system environment) connected to an IDS connector can range from a single AAS to all AASs present in the system environment. Across system environments, it is possible to establish connections from multiple connectors to a single connector endpoint.

The standardized AAS API also allows software engineers to easily develop additional applications, which access the asset and its data for various purposes such as process monitoring or dashboard applications. A process monitoring can also be realized by using various data processing applications, which use the AAS as data source and sink. For example, the aforementioned StreamPipes toolbox (see Section 3.1) can use the AAS as a data source and write processed events in an AAS property as AAS sinks. But also, the interoperability with other applications, such as a ML-based process monitoring, is facilitated by standardized interfaces and data structures. Furthermore, an AAS can also be used to encapsulate a ML model and describe its metadata, such as the data set used for training or model performance. This bears benefits when developing AI-based system, as models become more explainable and easier to exchange.

The combination with the IDS allows data from DTs to be protected by several usage policy classes [8]. In our case we protected the order data to be only viewable in the visualization app effectively cutting out misuse by forwarding customer details. The use of IDS components also led to operation of security-by-design software in the customer system environment, reducing the risk of data leaks and man-in-the-middle attacks. The IDS setup and integration however is additional effort which may only be worth it for highly critical data. Comprehension and effective application of the IDS infrastructure is not trivial and user friendliness requires improvements.

5 Related work

5.1 Use cases and applications of data-sovereign Industrie 4.0 Digital Twins

The combination of Industrie 4.0 Digital Twins with the IDS has also been tested in other projects, namely RI-OTANA by Fraunhofer ISST / IAIS [23] and the Smart Connected Supplier Network (SCCN) by TNO [24]. In both cases, the Digital Twin is either placed directly into the IDS connector (service container) or an IDS app is used to act as a proxy. Since the IDS connector only understands IDS messages, specific clients (other IDS connectors or web-based GUI) are necessary to communicate with the digital twin. On the plus side, usage policies can be effectively applied inside of the connector. However, partners without IDS background cannot access the digital twin resulting in a split: data-sovereign digital twins inside the IDS and digital twins without usage control.

We focus on keeping the digital twin outside of the IDS connector while extracting functionality in need of usage control inside of IDS apps. For example, critical production data inside of a Digital Twin is sent to an IDS app analyzing this data and can then be viewed by partners in an IDS visualization app. This combination means that we can still provide Digital Twins without usage control for partners without IDS knowledge, while also providing critical data for IDS apps processing it according to our usage rules. However, providing critical data to IDS apps needs additional effort in form of configuration of the IDS connector. Our future work will focus on bringing the IDS usage control directly into the Digital Twin by mapping and applying AAS usage rules into ODRL policies. For this, the existing ABAC specification inside the AAS Security specification would need an extension for usage control. This will also reduce the effort of configuration and the aforementioned split between data-sovereign and non-data-sovereign Digital Twins.

5.2 Further Asset Administration Shell implementations

In this section we present further implementations of the AAS specification. The implementations are chosen based on our subjective perception of their level of maturity and current or future relevance. This list does not claim to be complete.

Interoperability across implementations comprises an offline aspect, serializing an AAS using one implementation and deserializing it using another one, as well as an online aspect, exchanging a running AAS with another implementation while not changing the API of the AAS to the outside world, e. g., an application. Interoperability across the different implementations is very limited. For the offline aspect this is because the AAS meta model and de-/serialization rules are still subject to changes every few years and different implementations often implement different versions. For the online aspect this is because there is only a protocol-agnostic definition of the API and each implementation is doing its own mapping of this protocol-agnostic API to the protocols they support (which is mostly HTTP) resulting in slightly different HTTP APIs.

PyI40AAS is a Python SDK developed by the RWTH Aachen University, which at the current date implements a subset of the AAS features, such as the serialization and deserialization of JSON and XML [25]. However, interaction with the AAS using OPC UA or HTTP/REST interfaces is not supported yet. [26].

Eclipse BaSyx [27] is an open-source implementation of the AAS specification. It has been developed as part of the research projects BaSys 4.0, that ended in mid-2019, and its successor BaSys 4.2, running until mid-2022, funded by BMBF. Compared to BaSyx, our implementation provides multiple unique features such as the ubiquitous extendibility, integration with Apache StreamPipes as well as with the IDS.

The **AASXServer** [28] is a C# based server implemented by the Industrial Digital Twin Association (IDTA) [29] and its members to deploy AASX files. These files are DT packages consisting of XML files describing the DT and additional files like PDFs and images. In this aspect, the AASXServer operates similarly to FA³ST, that supports DT descriptions in JSON, XML, RDF, AutomationML and AASX. It also supports REST, MQTT and OPC UA endpoints offering the AAS API over these protocols. The endpoints in the AASXServer are not synchronized, effectively operating several different DTs for the same asset over different protocols. FA³ST therefore implemented synchronization of all endpoints to enable operating on the same DT with different protocols. Set-up effort of both solutions is low by providing docker containerization. Additionally, FA³ST also supports connections to assets which do not provide I4.0-compliant protocols.

NOVAAS [30] is a reference implementation of the AAS specification from the research institute Instituto de Desenvolvimento de Novas Tecnologias (UniNova). It has been developed in the context of the H2020 PROPHESY project which aimed to enhance predictive maintenance services in the industrial environment [31]. NOVAAS is based on Node-RED which is a flow-based programming tool and provides a graphical user interface as well as web-based interfaces for experienced end users. In contrast to FA³ST NOVAAS is not implemented as a typical code library but rather realized using only Node-RED which might be harder to integrate in other systems.

6 Conclusion and future work

We are now entering a phase of digitization where the use of DTs is rapidly increasing. Furthermore, DT tends to become more complex than before. An important part of this process is the emphasis on interoperability, which requires a paradigm shift from proprietary DTs to standardized DTs in open service and data sharing infrastructures. To accelerate the adoption of DTs and facilitate the DT creation, processing and integration, multiple standards should be considered. Indeed, the success and usefulness of DTs depend heavily on a standardized model and on standardized interfaces to interact with the DTs. To ensure reusability and extensibility of the DT components, it is necessary to standardize the building blocks of a DT. Finally, data sovereignty standards are crucial for the careful management of the DT sensitive information across enterprise boundaries to ensure the confidentiality of the information exchanged and to avoid misuse.

In this paper, we propose a new approach for realizing Industrie 4.0-compliant and data-sovereign DTs. The proposed approach is based on the use of multiple standards. We have developed the FA³ST software ecosystem that allows easy and fast creation and management of DTs according to the AAS specification. The FA³ST service components are aligned with the ISO 23247 DT reference architecture for manufacturing. These two standards ensure syntactical and semantic interoperability among the DTs and within the architecture of DT, respectively. For use and integration of DTs across company borders, we have integrated FA³ST with the trusted factory IDS connector to ensure data sovereignty. Initiatives such as GAIA-X and the International Data Spaces Association (IDSA) collaborate

to enable data-sovereign Digital Twins. We focused mainly on the IDS as GAIA-X is still in specification while the IDSA already provides implementations. However, we believe that IDS technology can be adapted for a future use within GAIA-X.

We are planning to add more functionality to FA³ST, e. g., publish/subscribe-based endpoints and useful caching algorithms. FA³ST will be published as open-source in the near future. Considering that FA³ST is AAS-compliant, it could make a significant contribution to the IDTA, which aims to drive the broad use and application of AASs based on open-source projects.

In the context of the CC-KING project [32, 33], FA³ST is used to create DTs for sorting processes to improve and adapt these processes with regard to the sorting quality. This will be achieved by applying DTs as a tool for the engineering of AI-based systems according to the **P**rocess model for **AI S**ystems **E**ngineering (PAISE). Structured data through standardized interfaces, DTs are beneficial for the engineering of robust and resilient AI-based systems. Further applications of FA³ST are discussed in the IDS-Industrial use cases, e. g., when considering the applicability of the AAS and IDS concepts in the context of open supply chain management and marketplaces for industrial production such as the Smart Factory Web [34].

Funding: This work was supported by the Competence Center Karlsruhe for AI Systems Engineering (CC-KING, https://www.ai-engineering.eu) sponsored by the Ministry of Economic Affairs, Labour and Housing Baden-Württemberg.

References

1. W. Kritzinger, M. Karner, G. Traar, J. Henjes and W. Sihn, "Digital Twin in manufacturing: A categorical literature review and classification," *FAC-PapersOnLine*, vol. 51, no. 11, pp. 1016–1022, 2018.
2. International Organization for Standardization, "ISO/DIS 23247-1 Automation systems and integration — Digital Twin framework for manufacturing — Part 1: Overview and general principles".
3. O. Hillermeier, M. Punter, K. Schweichhart and T. Usländer, "Data Sovereignty - Critical Success Factor for the Manufcaturing Industry," [Online]. Available: https://internationaldataspaces.org/wp-content/uploads/dlm_uploads/IDSA-Position-Paper-Data-Sovereignty%E2%80%93Critical-Success-Factor-for-the-Manufacturing-Industry.pdf [Accessed 04 05 2021].
4. Plattform Industrie 4.0, "Details of the Asset Administration Shell: From Idea to Implementation," [Online]. Available: https://www.plattform-i40.de/PI40/Redaktion/EN/Downloads/Publikation/vws-in-detail-presentation.pdf. [Accessed 04 05 2021].
5. DIN SPEC 91345:2016-04, "Reference Architecture Model Industrie 4.0 (RAMI4.0)," [Online]. Available: https://www.beuth.de/en/technical-rule/din-spec-91345/250940128. [Accessed 04 05 2021].
6. Plattform Industrie 4.0, "Details of the Asset Administration Shell - Part 1," [Online]. Available: https://www.plattform-i40.de/PI40/Redaktion/EN/Downloads/Publikation/Details_of_the_Asset_Administration_Shell_Part1_V3.pdf?__blob=publicationFile&v=5. [Accessed 04 05 2021].
7. Plattform Industrie 4.0, "Details of the Asset Administration Shell - Part 2," [Online]. Available: https://www.plattform-i40.de/PI40/Redaktion/EN/Downloads/Publikation/Details_of_the_Asset_Administration_Shell_Part2_V1.pdf?__blob=publicationFile&v=6. [Accessed 04 05 2021].
8. IDS Association, "IDS Reference Architecture," [Online]. Available: https://internationaldataspaces.org/wp-content/uploads/IDS-Reference-Architecture-Model-3.0-2019.pdf. [Accessed 06 05 2021].
9. A. Teuscher et al., "Anforderungen und Referenzarchitektur eines Security Gateways zum Austausch von Industriedaten und Diensten," Beuth-Verlag, 2020.
10. Fraunhofer IESE, "MYDATA Control," [Online]. Available: https://www.mydata-control.de/de/. [Accessed 12 05 2021].
11. International Data Spaces Association, "Trusted Connector User Documentation," [Online]. Available: https://industrial-data-space.github.io/trusted-connector-documentation/. [Accessed 06 08 2021].
12. International Data Spaces Association e.V., "Framework for the IDS Certification Scheme," 2019. [Online]. Available: https://internationaldataspaces.org/download/16416/. [Accessed 19 07 2021].
13. M. Jacoby, B. Jovicic, L. Stojanovic and N. Stojanovic, "An Approach for Realizing Hybrid Digital Twins using Asset Administration Shells and Apache StreamPipes," *Information*, 2021.
14. VDI, "VDI/VDE 2193 Blatt 1: Language for I4.0 components," 2020. [Online]. Available: https://www.vdi.de/richtlinien/details/vdivde-2193-blatt-1-sprache-fuer-i40-komponenten-struktur-von-nachrichten. [Accessed 28 07 2021].
15. Plattform Industrie 4.0, "I4.0-Sprache (German)," [Online]. Available: https://www.plattform-i40.de/PI40/Redaktion/DE/Downloads/Publikation/hm-2018-sprache.pdf?__blob=publicationFile&v=6. [Accessed 04 05 2021].
16. DIN SPEC 16593-1, "RM-SA – Reference Model for Industrie 4.0 Service Architectures – Part 1: Basic Concepts of an Interaction-based Architecture (Usländer, T. (Ed.)," 2018. [Online]. Available: https://www.beuth.de/en/technical-rule/din-spec-16593-1/287632675. [Accessed 06 08 2021]).
17. "Apache StreamPipes," [Online]. Available: https://streampipes.apache.org/. [Accessed 04 05 2021].
18. International Organization for Standardization, "ISO/DIS 23247-2 Automation systems and integration — Digital Twin framework for manufacturing — Part 2: Reference architecture".
19. International Data Spaces Association, "IDSA Use-Cases Overview," [Online]. Available: https://internationaldataspaces.org/make/use-cases-overview/. [Accessed 06 05 2021].

20. International Data Spaces Association e.V., "IDS Connector SwaggerHub API Documentation," [Online]. Available: https://app.swaggerhub.com/apis/idsa/ids-connector/. [Accessed 06 08 2021].
21. G. Maier, F. Pfaff, C. Pieper, R. Gruna, B. Noack, H. Kruggel-Emden, T. Längle, U. D. Hanebeck, S. Wirtz, V. Scherer and J. Beyerer, "Experimental Evaluation of a Novel Sensor-Based Sorting Approach Featuring Predictive Real-Time Multiobject Tracking," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 2, pp. 1548–1559, 2021.
22. OPC Foundation, "I4AAS – Industrie 4.0 Asset Administration Shell," [Online]. Available: https://opcfoundation.org/markets-collaboration/i4aas/. [Accessed 12 05 2021].
23. H. Haße, H. van der Valk, N. Weissenberg and B. Otto, "Shared Digital Twins: Data Sovereignty in Logistics Networks," in *Conference: New Ways of Creating Value in Supply Chains and Logistics - Hamburg International Conference of Logistics (HICL) 2020*, Hamburg, 2020.
24. International Data Spaces Association, "Smart Connected Supplier Network," [Online]. Available: https://internationaldataspaces.org/usecases/smart-connected-supplier-network/. [Accessed 10 05 2021].
25. T. Miny, M. Thies, U. Epple and C. Diedrich, "Model Transformation for Asset Administration Shells," in *IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society*, 2020.
26. RWTH AAchen, "GitLab of the RWTH Aachen University," [Online]. Available: https://git.rwth-aachen.de/acplt/pyi40aas. [Accessed 04 05 2021].
27. "Eclipse BaSyx," [Online]. Available: https://www.eclipse.org/basyx/. [Accessed 03 05 2021].
28. PHOENIX CONTACT GmbH & Co. KG, Festo SE & Co. KG, Fraunhofer IOSB-INA Lemgo, "AASX Server" [Online]. Available: https://github.com/admin-shell-io/aasx-server. [Accessed 07 05 2021].
29. Industrial Digital Twin Association e.V., "IDTA - Industrial Digital Twin Association e.V. Webseite," [Online]. Available: https://idtwin.org/. [Accessed 06 08 2021].
30. G. Di Orio, P. Maló and J. Barata, "Novaas: A reference implementation of industrie4. 0 asset administration shell with best-of-breed practices from it engineering," in *IECON 2019-45th Annual Conference of the IEEE Industrial Electronics Society*, 2019.
31. "PROPHESY," [Online]. Available: https://www.prophesy.eu/. [Accessed 10 05 2021].
32. Competence Center KI-Engineering (CC-KING), "KI-Engineering - AI System Engineering - The Karlsruhe Way," [Online]. Available: https://www.ki-engineering.eu/content/dam/iosb/ki-engineering/downloads/Definition%20KI-Engineering_OnePage_eng_web.pdf. [Accessed 06 05 2021].
33. Competence Center Karlsruhe for AI Systems Engineering, "CC-KING Website," [Online]. Available: https://www.ai-engineering.eu/. [Accessed 22 07 2021].
34. T. Usländer, F. Schöppenthau, B. Schnebel, S. Heymann, L. Stojanovic, K. Watson, S. Nam and S. Morinaga, "Smart Factory Web—A Blueprint Architecture for Open Marketplaces for Industrial Production," *MDPI Applied Sciences*, 2021.

Bionotes

MSc. Michael Jacoby
Fraunhofer IOSB, Fraunhoferstr. 1, 76131 Karlsruhe, Germany,
Phone +49 (0) 721 6091 470
michael.jacoby@iosb.fraunhofer.de

Research associate, Group "Modeling and System Networking", Department "Information Management and Production Control" at Fraunhofer IOSB, Karlsruhe; Digital Twins, Industrie 4.0 Asset Administration Shell, Internet of Things and semantic interoperability.

MSc. Friedrich Volz
Fraunhofer IOSB, Fraunhoferstr. 1, 76131 Karlsruhe, Germany,
Phone +49 (0) 721 6091 392
friedrich.volz@iosb.fraunhofer.de

Research associate, Group "Smart Factory Systems", Department "Information Management and Production Control" at Fraunhofer IOSB, Karlsruhe; industrial communication systems, data management, IT automation with focus on standardization, interoperability and security.

MSc. Christian Weißenbacher
Fraunhofer IOSB, Fraunhoferstr. 1, 76131 Karlsruhe, Germany,
Phone +49 (0) 721 6091 531
christian.weissenbacher@iosb.fraunhofer.de

Research associate, Group "Smart Factory Systems", Department "Information Management and Production Control" at Fraunhofer IOSB, Karlsruhe; Digital Twin applications and interfaces for AI systems.

Dr. Ljiljana Stojanovic
Fraunhofer IOSB, Fraunhoferstr. 1, 76131 Karlsruhe, Germany,
Phone +49 (0) 721 6091 287
ljiljana.stojanovic@iosb.fraunhofer.de

Head of Research Group “Smart Factory Systems”, Department “Information Management and Production Control” at the Fraunhofer IOSB, Karlsruhe; research, publications and standardization contributions on semantic technologies, complex event processing, Digital Twins and Industrie 4.0 models and systems.

Dr.-Ing. Thomas Usländer
Fraunhofer IOSB, Fraunhoferstr. 1, 76131 Karlsruhe, Germany,
Phone +49 (0) 721 6091 480
thomas.uslaender@iosb.fraunhofer.de

Head of Department “Information Management and Production Control” at Fraunhofer IOSB, Karlsruhe; research, publications and standardization contributions on agile service engineering, Industrie 4.0 architectural models, open geospatial service architectures and AI systems engineering.