M. A. Iñigo et al., "Towards Standardized Manufacturing as a Service through Asset Administration Shell and International Data Spaces Connectors," IECON 2022 – 48th Annual Conference of the IEEE Industrial Electronics Society, 2022, pp. 1-6.

DOI: https://doi.org/10.1109/10.1109/IECON49645.2022.9968592

Towards Standardized Manufacturing as a Service through Asset Administration Shell and International Data Spaces Connectors

Miguel A. Iñigo
Innovation and Technology dept.
MONDRAGON S.Coop.
Arrasate / Mondragón, Spain
Email: minigo@mondragoncorporation.com

Jon Legaristi
Felix Larrinaga
Electronics and Computer Science dept.
Mondragon University
Arrasate / Mondragón, Spain
Email: jon.legaristi@alumni.mondragon.edu
Email: flarrinaga@mondragon.edu

Alain Perez
Javier Cuenca
Electronics and Computer Science dept.
Mondragon University
Arrasate / Mondragón, Spain
Email: aperez@mondragon.edu
Email: jcuenca@mondragon.edu

Blanca Kremer
ICT dept.
Ikerlan Research Centre, BRTA.
Arrasate / Mondragón, Spain
Email: bkremer@ikerlan.es

Elena Montejo
Alain Porto
ICT and automation dept.
IDEKO Research Centre, BRTA.
Elgoibar, Spain
Email: emontejo@ideko.es
Email: aporto@ideko.es

***Abstract*—This paper presents an industrial scenario that simulates a Manufacturing as a Service system for the execution of remote production orders built upon the implementation of emerging Asset Administration Shell (AAS) capabilities and International Data Space connectors. Static and dynamic information from industrial assets (presses and laser cutting machines) are modelled with new AAS submodels and the result is stored in an AAS manager/registration system. A manufacturing orchestrator discovers assets through the registry and completes production orders. The AAS registry allows the selection of assets with capabilities to perform tasks and also shares the AAS catalogue available in the system. The catalogue is shared with external parties through Data Space Connectors. Third party companies can launch manufacturing orders remotely using the same connectors. The paper validates the implementation of AAS components and IDS connectors in a manufacturing context where remote production orders can be securely activated.**

***Index Terms*—Standardization, Interoperability, Asset Administration Shell, Manufacturing Data Spaces**

I. Introduction

Markets and new business models impose strong / changing requirements for industrial companies. Namely, highly customized products, high quality services and reduced prices [1]. To make the new industry paradigm (Industry 4.0) a success, companies need to implement more complex, interoperable, reconfigurable and responsive systems for their products and services. Cyber-Physical Systems (CPS) enable self-adaptation, self-configuration and self-diagnosis of industrial machinery, as well as covering inter/intra-enterprise integration for ubiquitous environments. Especially in manufacturing, CPSs enable migration from the existing hierarchical control structures based on the ISA 95 automation pyramid to more decentralized and reconfigurable systems that will be needed in the future. Changing requirements demand the digitization of industrial assets through the development of Information and Communication Technologies (ICT) that enable the coexistence of Operation Technologies (OT) of the manufacturing world with Information Technologies (IT) of the Internet.

Integration, interoperability and data management are the main drivers for this new industrial context. The lack of interoperability between heterogeneous systems is one of the main challenges hindering the great potential and the huge expectations placed around the integration of IoT and CPS solutions in the industrial context [2]. To meet the requirements of Industry 4.0, industrial products, services and processes must comply with reference models, standards and the I4.0 language. RAMI 4.0 [3], IIRA or NIST SME are the most relevant Smart Manufacturing Reference Models. Standard Development Organizations (SDOs), industrial organizations, joint committees and working groups work for the digitization of industry and develop standards for smart manufacturing such as IEC/ISO 62264 Enterprise - Control System Integration, IEC 62541 OPC Unified Architecture or IEC 62714 Automation Markup Language. Among the different initiatives, the Asset Administration Shell (AAS) proposed by the German platform Industrie 4.0 aims to describe an asset electronically in a standardized manner enabling interoperability between the different assets of a plant.

In addition to standardisation, one of the key issues in the I4.0 context is data management. The evolution of industrial organisations in recent years has created a set of challenges when it comes to managing data. The collection and exchange of data between different industrial spaces is a challenge. The growth of this data exchange between companies can generate added value and accelerate innovation. As explained in the White Paper [4], data sharing generates added value in the following fields:

- Enhancing asset optimization. By combining data from multiple users in the same industrial environment allowing manufacturers to implement algorithms that are used to improve maintenance.
- Tracking products along the value chain enabling manufacturers to improve their production planning, reduce inventory levels and react faster to unexpected events in the supply chain.
- Tracing process conditions along the value chain. Manufacturers have access to a continuous and complete digital record throughout the value chain ensuring that suppliers follow agreed production processes.
- Exchanging digital product characteristics. Sharing data of digital product twins allows manufacturers to optimize connected production processes.
- Verifying provenance. To meet customer demands, manufacturers need transparency regarding the processing and authenticity of their supplies.

According to [5], for the correct exchange of data between industrial sites or companies it is essential to consider the following four pillars: interoperability, trust, data value and data governance. The International Data Spaces Association (IDSA)[1] proposes a secure and sovereign system of data exchange that complies with those pillars. The so-called Industrial Data Spaces (IDS) aims to get maximum value from data.

Considering these standard components, the main objective of this paper is to build an industrial demonstrator using the latest developments of the AAS and IDS initiatives. The paper is structured as follows. Section II provides the background behind the standards addressed in the paper. Section III presents the motivation for this work. Section IV, addresses the implementation of the proposal in a use case. Finally, Section V presents the conclusions

II. Background

The AAS initiative is based on IEC 62832 - Digital Factory Framework and defined in IEC 63278 to support the idea of standardized automated industrial systems, industrial assets and CPSs throughout the manufacturing lifecycle within a digital environment where the digital twin approach is feasible. The AAS initiative plays a relevant role for further developments in the Industry 4.0 landscape and Language I4.0. Recently there has been significant progress in relation to AAS specifications working towards an Industry4.0 platform:

1) New versions of the AAS metamodels [6] have been generated by improving and modifying the original aspects described in the metamodel elements.
2) The first version of the API specification [7] to allow access to the information provided by an AAS has been presented.
3) Work has started on the specification of sub-models [2]. The nameplate sub-model and the technical data sub-model have been specified.

In addition, the University of Magdeburg has proposed libraries and tools to implement new AAS scenarios[3]:

1) Type 1, also called passive AAS, maps the metamodel to a specific file format (aasx). This allows the AAS to be interchangeable as a digital catalogue between suppliers and integrators. This catalogue will contain and standardise the information that is usually distributed in different data sources.
2) Type 2, also called reactive, defines an interface for computationally capable nodes to provide information about the assets they control or access.
3) Type 3 or proactive. In this case, AAS can interact with each other by having a common representation model, but this requires a messaging specification to enable this communication. There is also a specification for an I4.0 language to enable this scenario. These AAS are known as I4.0 components.

With this approach, the concept of *Active AAS* is defined. This type of AAS can interact with other AAS using a common representation model. A messaging specification has also been developed to enable communication between active AASs. Active AASs communicates using the specification to perform more complex tasks. For example, specialized assets (manufacturing order orchestrators) will be able to read manufacturing orders and contact other assets to perform the tasks required for the completion of those orders (processes).

A final consideration concerns the management of a large number of AAS. We need a system to enables the registration and discovery of AAS on a production plant, so that one asset can identify, select and communicate with another asset based on specific requirements. The Asset Administration Shell Package Explorer seems to provide the necessary components to enable AAS registration and management in an industrial plant [8].

The IDS initiative proposed by the IDSA aims to build an ecosystem that facilitates data exchange in a ”secure, trusted and semantically interoperable way” between business partners, which ensures the sovereignty of the data offered by the provider at all times through a series of rules and usage policies [9]. The IDS Reference Architecture Model (IDS-RAM) [10] describes the ontology [9] and the key components that are necessary within a sovereign data ecosystem in order to fulfil the above-mentioned characteristics during

[1] https://internationaldataspaces.org/

[2] https://www.sci40.com/interopera/

[3] https://www.i40.ovgu.de/i40/en/Asset+Administration+Shell/Meta+Model.html

the exchange of data between two stakeholders. The general structure of the IDS-RAM is composed of five layers [10]:

- The Business Layer. Defines the roles and the interactions between roles of the IDS participants
- The Functional Layer. Defines the functional requirements of the IDS.
- The Process Layer. Specifies the interactions between components of the IDS ecosystem.
- The Information Layer. Specifies the Information Model, facilitating compatibility and interoperability.
- The System Layer. Defines aspects as integration, configuration and deployment of software components.

The core of the IDS data ecosystem is the IDS Connector. This component is in charge of exchanging all kinds of information (data sets, metadata, etc.) between the participants in the data space. In addition, it is the component that ensures the secure and trusted exchange of data.

There are several implementations of the IDS Connector that meet the requirements defined in the IDS Reference Architecture Model. One of the most advanced and currently used in projects working in the context of Industry 4.0 such as Mobility Data Space[4], the Energy Data Space [11], and the Bauhaus.MobilityLab[5] is the Dataspace Connector (DSC) developed by Fraunhofer Institute for Software and Systems Engineering (ISST). The connector is open source software and is available on IDSA's GitHub repository[6].

III. Motivation

These recent advances in the AAS specifications and the growth of the IDS initiative make an invaluable contribution to the integration and interoperability between industrial devices of heterogeneous systems for collaborative industrial production environments. In this I4.0 context, Mondragon Corporation participates in several initiatives to boost the digitalization and interoperability among the cooperatives in its industrial group. This collaboration is materialized through digital transformation European projects (Arrowhead[7], Mantis[8], Productive 4.0[9] and QU4LITY[10]). Following in the footsteps of the first AAS prototype developed by MONDRAGON Corporation for the digital transformation of its industrial environment [12], the aim of this paper is to develop an industrial context in which an Active AAS supported by the new AAS sub-models and a AAS manager and registry system is implemented. In addition, a system of IDS connectors is to be implemented to allow secure and sovereign data exchange between the mentioned AAS management system and external business partners. The motivation is to provide experiences to Mondragon industrial companies and move towards a standardised manufacturing mode of operation.

[4]https://mobility-dataspace.eu/

[5]https://bauhausmobilitylab.de/

[6]https://github.com/International-Data-Spaces-Association/DataspaceConnector

[7]https://www.arrowhead.eu/

[8]https://industry4e.eu/project/mantis/

[9]https://productive40.eu/

[10]https://qu4lity-project.eu/

IV. Use case

The use case proposed in this paper consists of an industrial scenario that simulates a Manufacturing as a Service system. The demonstrator considers stamping and laser cutting processes to produce an oven door and a refrigerator door using 2 manufacturing orders:

- The oven door manufacturing process consists of 3 tasks: (1) a 2kW power laser cutting machine, (2) a 2,000kN pressing machine and (3) another 8kW laser cutting machine.
- The refrigerator door manufacturing process consists of 2 tasks: (1) involving a 3,000kN press machine and (2) an 8kW laser cutting machine.

The demonstrator is based on an AAS architecture in which the plant assets (in this case presses and laser cutting machines) register their AAS in an AAS Manager/Register system. A software component, the orchestrator, in charge of manufacturing orders, queries the AAS Manager/Register system to identify assets capable of executing a given work order. The orchestrator extracts from the AAS responses received, the protocol and endpoint to interact with the appropriate assets. Once communication is established, the orchestrator interrogates the assets on their availability and, if possible, orders the execution of the tasks foreseen in the production order. Finally, through the IDS Connectors system, the option of executing manufacturing orders from an external system is enabled. The supplier offers the catalogue of every AAS registered in the system so that, once the available resources and services are known, the third party is able to launch a manufacturing order from its connector.

The architecture of the use case is shown in Figure 1 and considers the following requirements:

- 4 industrial assets (2 industrial presses and 2 laser cutting machines) with different capabilities.
- Execute production orders received in JSON format.
- Integrated management of all AASs available in a plant together with their capabilities through the representation model.
- Execution of the AASs through an OPC-UA server for each of the AASs
- Orchestrator of the solution taking into account production needs with industrial capabilities at plant level and implementation.
- 2 Dataspace Connectors, one part of the company providing the manufacturing services, and the other part of the company that wants to consume these services.
- The manufacturing orders can be executed through an exchange between the IDS Connectors.
- Consumption of the catalogue of AASs available through the IDS ecosystem.

The details of the main components in the use case are presented next.

A. Definition of AAS models for presses and laser machines

An AAS has been defined for each of the industrial assets involved in the demonstrator. The assets, through their AAS,

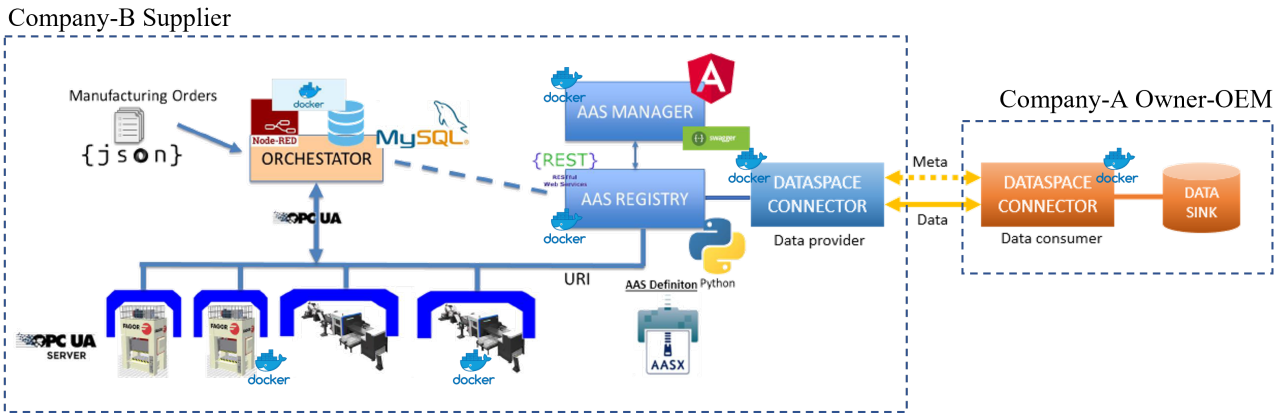

Fig. 1. Architecture of the use case.

will show both static information (fixed machine characteristics) and dynamic information (data that is modified during machine operation) grouped in different sub-models.

Nameplate and TechnicalData submodels: defined by the Platform4.0 initiative and collected in the specification [13] and [14]. Nameplate provides information describing the asset identification, properties such as manufacturer name, product family and serial number are collected in this sub-model. TechnicalData contains information about the technical characteristics of the machine deployed in the plant, in our case the maximum nominal force of the machines has been included.

ProductionData and OpParamSettings submodels: custom sub-models defined for the use case. ProductionData sub-model allows grouping the variables that provide information about the operational status of the machine (available, running or out of service) and information about the number of pieces manufactured since the last die change in the case of presses. OpParamSettings sub-model contains the externally accessible machine operations; in our demonstrator, Start and EmergencyStop are provided by the machine's AAS.

M2MConnectivity submodel: custom sub-model defined to ease the communication with other AASs (e.g. Orchestrator). This sub-model contains information on the protocols handled by the machine. Through these protocols the AAS will provide the data of the dynamic sub-models. Each protocol will be uniquely represented by a semanticId and the connection parameters will be defined in one or more properties.

Each machine, at start-up, on the one hand, will register its AAS, with the sub-models containing the static information, in the plant's AAS manager and on the other hand, will raise an OPC-UA server that will provide the information of the dynamic sub-models.

B. AAS Manager and Registry System

The AAS registry is a common storage point for all AASs of a given workshop, company or manufacturer. In this case, a registry is used to store the passive AASs of the use case, so that they can be accessed by third party services that can perform any management on the assets. The registry has been created using a REST API implemented in python. The registry also performs an AAS validation and stores it in a database (specifically in CouchDB). In addition, it offers the possibility of filtering by wrapping the database searching engine in the REST API interface.

On the other hand, to facilitate the management of the registry, an additional tool, the AAS manager, has been included. The Manager is a user interface in the form of a web application that allows a non-expert user to manage the entire AAS registry and to explore the different registered assets.

C. Active AAS (Orchestrator)

The orchestrator is the component of the use case that is responsible for receiving manufacturing orders and deciding which asset will execute the different tasks in the order. The functionality of the orchestrator is summarised below:

- Dynamically manages production orders.
- Queries the AAS registry for assets that satisfy each of the tasks in an order.
- It relies on production order requirements to select the most appropriate assets.
- Dialogues with the assets to check their availability.
- It orders the start-up of each of the assets to satisfy the achievement of the tasks of an order.

The orchestrator data flow is defined using Node-RED flows. These allow us to develop the logic in a visual manner using modular nodes that offer different functionalities, such as communicating via OPC-UA, making REST calls or accessing a database. Thanks to the deployed AAS Registry, the communication with the assets is dynamic, allowing the orchestrator to decide the right asset to perform a task and to know if that asset is available.

Communications: The orchestrator talks to the other components of the use case through different protocols:

- Client: the client is the actor that would generate a production order. For this, the orchestrator communicates with the Dataspace Connector through a REST API provided by the orchestrator itself where an order can be sent in a JSON file.

- AAS Registry: the orchestrator communicates with the AAS registry via REST to collect the list of available machines in the manufacturing plant.
- Machines (or assets): the machines in this use case communicate via OPC-UA, providing the information for this communication within the M2MConnectivity model within its AAS description (AASX file registered in the AAS Registry).

Data Models: The orchestrator needs to know the processes to manufacture the different products, as well as to store the information of the orders received and the steps carried out. For this purpose, the following data model has been developed in this use case. The tables of the model and their purpose are described below:

- Order: each time a customer POSTs an order, a new record is added to this table. Its status is updated during the production of the order.
- Ordered Product: this table adds a line for each product in an order and is updated each time a product is manufactured.
- Process: This table stores the process to be performed for the manufacture of a product.
- Task: This table stores the tasks of a process, i.e. the tasks to be performed to manufacture a product.
- Task Property: this table stores the requirements that a task has in order to be executed. For example, the force that a press needs to executed the task. The value of this table will be used to filter the machines capable of performing a task. The semantic id field will allow us to search for a property in the AAS sub-models in an unambiguous way.

Operation: The orchestrator starts working when it receives a request from the Dataspace Connector to its REST service with the manufacturing order requested by the client. The orchestrator stores the request in the database and start the manufacturing process sequentially for each product in the order. First, it loads the tasks and properties of the manufacturing process from the database. It continues by requesting the available sub-models and assets in the AAS Registry to filter those that meet the properties of the tasks to be performed. Once these assets have been filtered, it communicates with them. Finally, the status of the order is updated every time a task, a process or the whole order is finished.

Whenever the orchestrator has to communicate with an asset, it needs to know how to do so. The AAS definition of each machine stores the communication information within a M2MCommunication sub-model. Thus, the orchestrator asks the AAS Registry for the asset's M2MCommunication sub-model, extracting the information needed for communication (e.g. the OPC-UA endpoint). Once the orchestrator knows how to communicate with the asset, it contacts the asset directly asking for its status. If the asset is not available, the orchestrator attempts to communicate with another asset in its list. If the asset is available, it assigns the task to the asset and starts it up. It then extracts the status of the asset until the machine has finished the task. Finally, when the task is finished, the communication with the asset ends and the next task is invoked or the order is finished.

D. IDS Connectors

The IDS Connectors are responsible for the exchange of data and services between the different systems or companies. Since the simulated system works on the basis of a Manufacturing as a Service model, the connectors are responsible for the following two main aspects:

- Sharing the AAS catalogue available in the system. The client can request the list of AASs registered in the supplier system via the Dataspace Connector in order to visualise and analyse the available resources.
- Launch the manufacturing order in the orchestrator. The client launches the manufacturing order through the connector which, via the subscription system, handles the request and executes the recipe in the orchestrator.

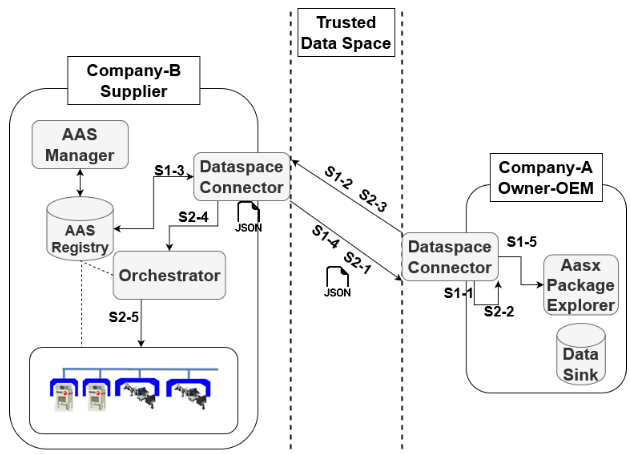

Fig. 2. DSC operation sequence diagram.

Data Model: The data model of the Dataspace Connector is based on the structure defined in the IDS Infomodel as explained on the official DSC website[11]. Basically, it consists of a series of linked resources that, through the sum of the defined metadata, describe the resource to be offered. These are the elements that make up the data model: Resource, Catalog, Representation, Artifact, Agreement, Rule and Contract.

Operation: to be able to consume the AAS catalogue from the client side, resources following the data model explained before must be created in the supplier connector. Then, from the customer's connector, the contract agreement must be made in order to consume the resource with the catalogue. Once this process of resource creation has been completed, the process that the system performs to exchange the catalogue is explained in Figure 2 as *Sequence 1*. The client (Company-A Owner-OEM) requests the catalogue (S1-1), this request is managed in its local connector, which requests the resource that contains it from the connector of the supplier company

[11]https://international-data-spaces-association.github.io/DataspaceConnector/

(Company-B Supplier) (S1-2). From the supplier system, the AAS Registry is accessed and the resource is updated with the list of stored AASs (S1-3). Finally, the catalogue is sent back by the connector in JSON format (S1-4) and the client visualises and analyses it through the AASX Package Explorer (S1-5).

For the manufacturing order execution process, it is necessary, firstly, to create the resources in the customer connector (Company-A) following the IDS data model, and secondly, make the contract agreement from the supplier connector (Company-B) in order to consume the resource with the order. After these steps the orders can be requested. The whole process is explained in Figure 2 as *Sequence 2*. First Company-B connector subscribes with Company-A connector (S2-1) in order to be notified when a new order is placed there. When the customer (Company-A Owner-OEM) updates the mutually agreed resource (S2-2) with the production order, the subscription system notifies and sends the recipe to be executed to the supplier's connector (S2-3) which forwards the received order to the orchestrator (S2-4) in order to start the manufacturing sequence (S2-5).

V. Conclusions and Future Work

In response to the motivation identified in section III, we have developed and tested a use case that simulates a Manufacturing as a Service system where a customer and a supplier exchange production data according to standards. The current state of the AAS and Manufacturing Data Spaces Connectors have been validated by means of a use case that implements 4 industrial assets (2 presses and 2 laser cutting machines) and an orchestrator that manufactures orders for 2 products (oven and refrigerator doors). Emphasis has been put on the management of AAS at plant level, managing the flow of information for the execution of manufacturing orders from an external company. The feasibility of the technology has been demonstrated to represent not only heterogeneous industrial assets and their digital twin, but also to enable interoperability between companies in a manufacturing supplier to OEM context. To achieve this use case we have deployed open-source components provided by the AAS and IDS community and develop adapters to integrate them in a platform. All components are offered in Docker technology that makes the solution highly scalable and easy to replicate.

Although the Industry 4.0 Platform has made great efforts towards interoperability and standardized RAMI 4.0 digital twins, more AAS sub-models and their implementation and validation are necessary. A similar conclusion can be withdrawn for Manufacturing Data Spaces. Data Space Connectors are a promising solution to allow secure flow of information among industries and for the exchange of data in an industrial business service context. Nevertheless, further research and the development of data exchange scenarios are requested to cover all industrial requirements. For future developments and prototypes we will consider the integration of new AAS sub-models and IDS connectors to scale the solution to other industrial scenarios. Our next objective is to build a federated industrial environment at edge/cloud level to accommodate Artificial Intelligence and machine learning algorithms in concordance with the GAIA-X and CATENA-X initiatives.

References

[1] I. Veza, M. Mladineo, and N. Gjeldum, "Managing innovative production network of smart factories," *IFAC-PapersOnLine*, vol. 48, no. 3, pp. 555–560, 2015, 15th IFAC Symposium onInformation Control Problems inManufacturing. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S240589631500378X

[2] E. Jantunen, G. Di Orio, C. Hegedűs, P. Varga, I. Moldován, F. Larrinaga, M. Becker, M. Albano, and P. Maló, "Maintenance 4.0 world of integrated information," in *Enterprise Interoperability VIII*, K. Popplewell, K.-D. Thoben, T. Knothe, and R. Poler, Eds. Cham: Springer International Publishing, 2019, pp. 67–78.

[3] M. Hankel and B. Rexroth, "The reference architectural model industrie 4.0 (rami 4.0)," *ZVEI*, 2015.

[4] F. Betti, F. Bezamat, M. Fendri, B. Fernandez, D. Küpper, and A. Okur, "Share to gain: Unlocking data value in manufacturing," *URL: http://www3. weforum. org/docs/WEF_Share_to_Gain_Report. pdf [Stand: 27.04. 2020]*, 2020.

[5] L. Nagel and D. Lycklama, "Design principles for data spaces," *Position Paper, International Data Spaces Association*, 2021.

[6] S. Bader, E. Barnstedt, H. Bedenbender, B. Berres, M. Billmann, B. Boss, A. Braunmandl, E. Clauer, C. Diedrich, B. Flubacher, W. Fritsche, K. Garrels, A. Gatterburg, M. Hankel, O. Hillermeier, M. Hoffmeister, M. Jochem, A. Köpke, Y. Kogan, and C. Ziesche, "Details of the asset administration shell. part 1 - the exchange of information between partners in the value chain of industrie 4.0 (version 3.0rc01)," Tech. Rep., 11 2020.

[7] S. Bader, B. Berres, B. Boss, A. Gatterburg, M. Hoffmeister, Y. Kogan, A. Köpke, M. Lieske, T. Miny, J. Neidig, A. Orzelski, S. Pollmeier, M. Sauer, D. Schel, T. Schröder, M. Thron, T. Usländer, J. Vialkowitsch, F. Vollmar, and C. Ziesche, "Details of the asset administration shell. part 2 -interoperability at runtime - exchanging information via application programming interfaces (version 1.0rc01)," Tech. Rep., 11 2020.

[8] C. Wagner, J. Grothoff, U. Epple, R. Drath, S. Malakuti, S. Grüner, M. Hoffmeister, and P. Zimermann, "The role of the industry 4.0 asset administration shell and the digital twin during the life cycle of a plant," in *2017 22nd IEEE international conference on emerging technologies and factory automation (ETFA)*. IEEE, 2017, pp. 1–8.

[9] S. Bader, J. Pullmann, C. Mader, S. Tramp, C. Quix, A. W. Müller, H. Akyürek, M. Böckmann, B. T. Imbusch, J. Lipp *et al.*, "The international data spaces information model–an ontology for sovereign exchange of digital content," in *International Semantic Web Conference*. Springer, 2020, pp. 176–192.

[10] B. Otto, S. Steinbuss, A. Teuscher, and S. Lohmann, "Ids reference architecture model," Apr 2019. [Online]. Available: https://zenodo.org/record/5105529

[11] V. Janev, M. E. Vidal, K. Endris, and D. Pujic, "Managing knowledge in energy data spaces," in *Companion Proceedings of the Web Conference 2021*, 2021, pp. 7–15.

[12] M. A. Iñigo, A. Porto, B. Kremer, A. Perez, F. Larrinaga, and J. Cuenca, "Towards an asset administration shell scenario: A use case for interoperability and standardization in industry 4.0," in *NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2020, pp. 1–6.

[13] W. G. . S. of the ZVEI and S. a. N. P. I. . Working Groups "Reference Architectures, "Submodel Templates of the Asset Administration Shell - ZVEI Digital Nameplate for industrial equipment (Version 1.0)," Federal Ministry for Economic Affairs and Energy (BMWi), Tech. Rep., 2020. [Online]. Available: https://www.plattform-i40.de/PI40/Redaktion/EN/Downloads/Publikation/Submodel_templates-Asset_Administration_Shell-digital_nameplate.html

[14] Working Group "Models and Standards" of the ZVEI and S. a. N. P. I. . Working Groups "Reference Architectures, "Submodel Templates of the Asset Administration Shell - Generic Frame for Technical Data for Industrial Equipment in Manufacturing (Version 1.1)," Federal Ministry for Economic Affairs and Energy (BMWi), Tech. Rep., 2020. [Online]. Available: https://www.plattform-i40.de/PI40/Redaktion/EN/Downloads/Publikation/Submodel_templates-Asset_Administration_Shell-Technical_Data.html