OPEN INDUSTRY 4.0 ALLIANCE

White Paper

AAS Tool Landscape

IMPRINT

Publisher
Open Industry 4.0 Alliance
Christoph Merian-Ring 12, 4153 Reinach, Switzerland
https://openindustry4.com
info@openindustry4.com

Status
File: AAS Tool Landscape
December 17th, 2025 – Version 1.1.0

Editor
Lucas Wolf (Open Industry 4.0 Alliance)

Lead Author
Dr. Daniel Lehner (TwinTech)

Co-Authors
Christian Methe (Twindustrial)
Moritz Hofer (XITASO)

Reviewers
Prof. Andreas Wortmann (ISW Stuttgart)
Christoph Röbl (Lenze Austria)
Jérôme Pfeiffer (ISW Stuttgart)
Prof. Judith Michael (Universität Regensburg)
Markus Kiele-Dunsche (Lenze)

Inhalt

1 Motivation 4

2 You Are Really My Type 5

2.1 AAS Type 1: File-Based Exchange 5

2.2 AAS Type 2: Networked Interface 6

2.3 AAS Type 3: Proactive AAS 6

3 AAS-Based Digitalization Infrastructure 8

3.1 The Architecture 8

3.2 How to realize digitalization use cases with an AAS-based infrastructure 10

3.3 Feels Overwhelming? Start Small, Scale Later! 12

4 Landscape of existing AAS Tools 14

4.1 Tool Overview 14

4.2 Further Comparison and Insights 15

5 Conclusion 18

1 Motivation

Introducing the Asset Administration Shell (AAS) into practice is not straightforward. In recent years, a wide range of tools and approaches has emerged, creating a fragmented landscape. One of the main drivers for using the AAS is the growing focus on data spaces. In Manufacturing-X, a BMWE funded initiative, Factory-X shows how companies can exchange data in a structured way. The MX-Port variants Leo, Hercules and Orion play a central role: Leo provides an AAS based entry into a Factory-X dataspace, Hercules combines Data Space Protocol with AAS interfaces, and Orion links OPC UA shop floor data to data spaces. Funding bodies therefore expect all X-projects to include the AAS in their architecture. DAVID-X was created to make this accessible for small and mid-sized companies.

This “wild growth” shows how much interest there is in the topic, but it also creates uncertainty: many solutions cover only parts of the AAS, some are limited to a specific type, and others focus on very narrow use cases. For companies, this raises a central question: where to begin, and which approach is future-proof? To create clarity, it is important to understand the different AAS types. Not every tool is able to implement all of them, and the maturity of these types varies significantly. In this white paper, we focus on Type 2, which at the time of publication represents the standard for scalable and interoperable AAS implementations. While Type 1 played an important role for early pilots and Type 3 is still in exploration, Type 2 provides the stable basis that can already be applied in real industrial environments. Not sure about the AAS or why it is so important? Read here.

Our aim is not only to explain what tools exist, but also to show how they interact as part of a broader architecture. Building on existing overviews such as the IDTA Solutions Hub[1], we conducted additional research, engaged with experts, and collected feedback from fairs and community events. At the same time, we want to lower the barrier to entry: it is possible to start small, with minimal investment, and still achieve first quick wins. From there, implementations can scale step by step, building towards more advanced use cases. To support this journey, we also share examples and best practices that illustrate how AAS is applied in concrete scenarios — because the variety of tools and concepts can feel overwhelming at first glance.

If you are unfamiliar with the AAS types, let us start there and guide you through the next chapter, which introduces the three types and explains their role.

[1] https://industrialdigitaltwin.org/solutions-hub

2 You Are Really My Type

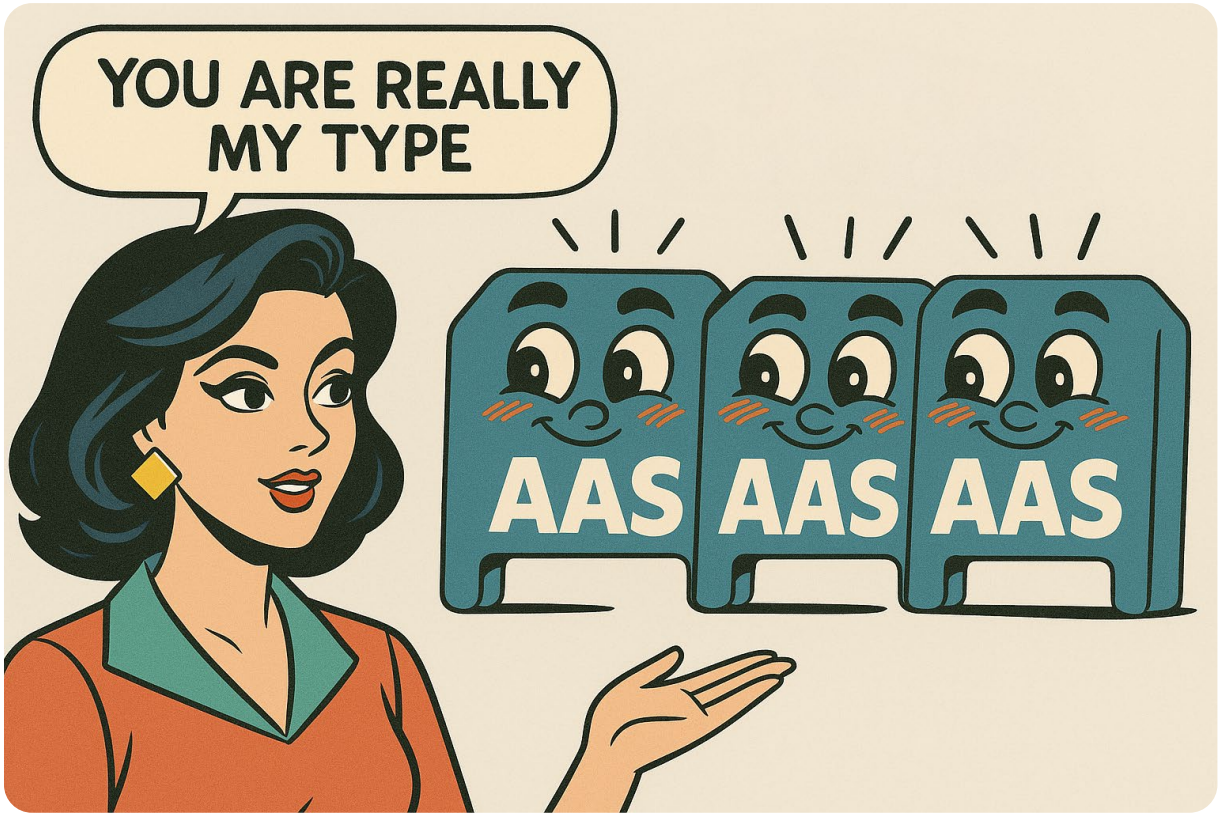


Figure 1: The 3 AAS types

2.1 AAS Type 1: File-Based Exchange

The first generation of AAS implementations is file-based. Here, the entire AAS is packaged into a file (e.g., an. aasx container) and exchanged manually or via file interfaces. This approach enabled early prototyping and provided a practical way to model and share information, but it lacks scalability. There is no real-time interaction, versioning is cumbersome, and the integration into IT/OT environments is limited.

However, there is already value that can be gained by the Type 1 AAS:

- Manufacturers can, for example, provide digital product information such as contact information or handover documentations a virtual extension of their physical products. This not only enhances customer experience but also supports compliance with upcoming regulatory requirements.

- Mechanical engineering companies can replace printed handover documentation with digital equivalents, as required by the new EU Machinery Regulation. Using a Type 1 AAS, this information can be delivered in a format that customers can

directly import into their own AAS tools. A digital nameplate can further simplify identification of the machine within the customer's toolchain, for instance in advanced planning or product lifecycle management systems.

2.2 AAS Type 2: Networked Interface

Type 2 represents the current standard and the focus of this work. Instead of static file exchange, the AAS is provided as a service with a defined API. This allows systems to access submodels dynamically, retrieve and update data, and connect assets directly to enterprise and shopfloor systems. Type 2 makes the AAS scalable, interoperable, and suitable for industry use. It is the recommended way forward, and many tools and frameworks already support this approach.

With this type 2 AAS, organizations can now:

- Provide the latest digital handover documentation or DPP directly to their customers, accessible through a QR code placed on the product or machine.

- Use AI-based tools such as predictive maintenance services to query information about the asset through the standardized AAS. This standardization enables a uniform connection between AI tools and assets, making it easy to scale use cases like predictive maintenance across the entire shop floor.

2.3 AAS Type 3: Proactive AAS

Type 3 is still in exploration and aims to extend the AAS from a passive data container to an active participant in the value chain of a product. Instead of only describing assets, the proactive AAS would also carry and execute rules, making it capable of triggering actions or calculations that proactively influence the behaviour of a system. In the context of manufacturing, this could lower integration costs, speed up adaptation to changes, and enable assets to "plug in" seamlessly when replaced or extended. If shared across companies, a proactive AAS could automate collaboration along the value chain and open the door to predictive services.

The type 3 AAS enables advanced, proactive use cases based on data and business logic stored within the AAS as standardized and interoperable format.

For example, organizations can:

- Calculate the storage location of a product during manufacturing based on dynamic process parameters. For example, if a quality check identifies an issue, it can request an available storage space in the defective area of the warehouse and route the product there. Once the defect storage is full, it can automatically trigger the next processing step, such as recycling.

- also automatically trigger warnings to operators or mitigate anomalies. If an AI service detects unusual machine behavior, it can schedule maintenance, or in case of a severe anomaly, temporarily shut down the machine to prevent damage or breakdowns.

With this understanding of the three AAS types, we note that in the following chapters, we focus on Type 2 AAS, as Type 3 is not yet standardized, and we see Type 1 AAS as prerequisite that is also included in Type 2. Our next step is to look at the architecture that shows how the individual building blocks work together to bring these models to life.

3 AAS-Based Digitalization Infrastructure

3.1 The Architecture

Behind every interoperable AAS use case (we described some of them above) lies an architecture that connects existing systems and assets, translates their data into a common structure, and makes it usable across applications. This chapter provides an overview of how these building blocks interact to turn heterogeneous information into standardized AAS models.

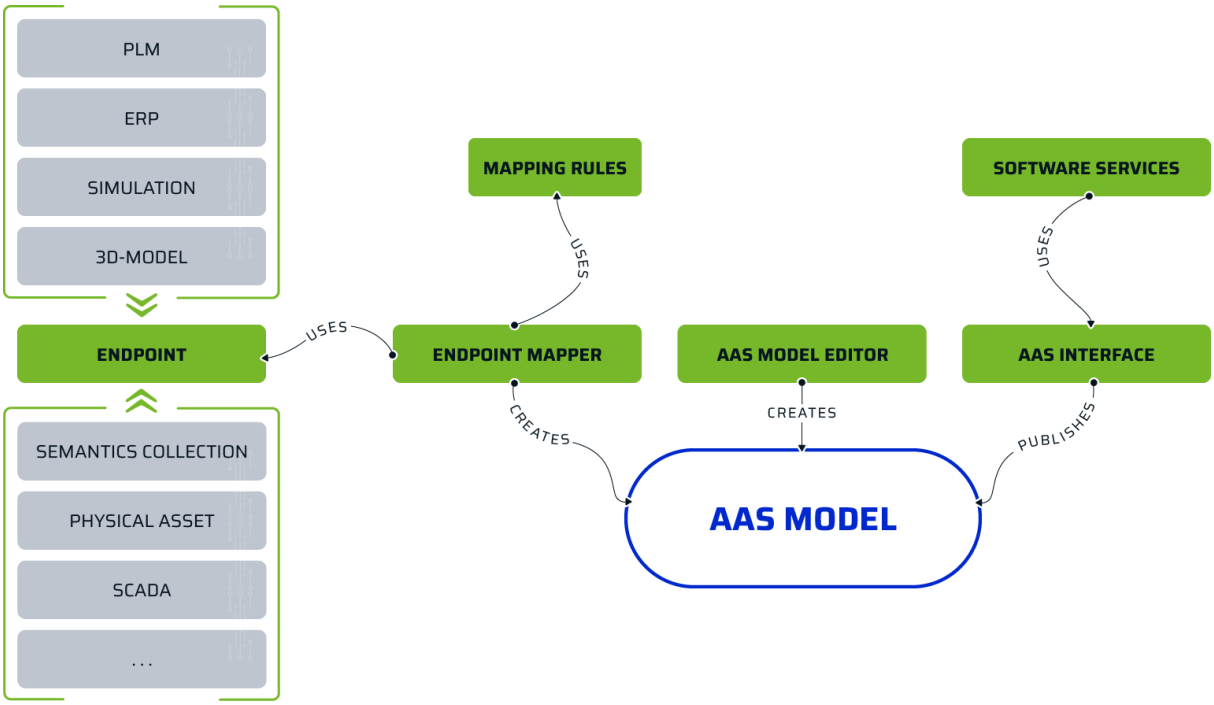


Figure 2: Overview of components of an AAS-based digitalization architecture

AAS Model: The central artifact of this architecture is the AAS model (usually available as .aasx or .json file). It contains information about available assets, data omitted by these assets, and available interaction mechanisms for these assets. It is defined using the AAS language, standardized by the Industrial Digital Twin Association (IDTA). The IDTA also provides a set of standardized templates (cf. submodel repository of the IDTA[2]) that can be reused for commonly representing data for specific purposes, such as digital nameplates, hierarchical structures (to, e.g., for a bill of materials), or product carbon footprints.

[2] https://industrialdigitaltwin.org/en/content-hub/submodels

AAS Model Editor: The model editor is a software tool that provides a user-friendly interface to define and maintain the structure of the AAS model. It creates the "blueprint" of the AAS model, describing which data and attributes should be represented and how they are organized. This provides the framework into which live data can be mapped.

Endpoint: The actual asset data is usually spread across multiple enterprise systems, such as PLM, ERP, MES, SCADA software, or the machines which regularly omit telemetry data. In the context of the AAS, these systems are referred to as Endpoints.

Endpoint Mapper: The endpoint mapper takes heterogeneous asset data from different endpoints and aligns it with the AAS model as a common format. It uses mapping rules to automatically enrich the AAS model with domain-specific data. By doing so, it acts as the translator between existing systems and the AAS world. In the AAS world, it is the fuel that brings the static AAS model, which is created through a model editor, to life.

Mapping Rules: Mapping rules define how data from a specific endpoint are transformed into the standardized structures used in the AAS model. They ensure consistency and interoperability, so that different systems can contribute information without losing meaning or structure. Often, they also allow the automated generation of AAS models after new assets are onboarded within an Endpoint (e.g., a new asset is created in the ERP after it was bought from a supplier).

AAS Interface: The AAS interface publishes the created AAS models via a standardized application programming interface (API), as defined by the IDTA[3], making them accessible for external use. Thus, the AAS interface is what differentiates type 2 from type 1 AAS models. It ensures that AAS models are available via the network in an interoperable way, and can be consumed by software services, applications, or even other organizations. In practice, this often includes an AAS Repository, which stores and manages the models and provides mechanisms for security and authentication. Usually, AAS repositories are decentralized data stores, i.e., each repository is owned by one individual organization within the value chain of a product. locate models for specific assets across multiple decentralized repositories, an AAS registry is used. In the case of the transfer of an asset between company borders (e.g., a manufacturer buys a component from a supplier) two options are available for transferring the respective AAS model of this asset from the supplier's to the manufacturer's own repository. Option 1: the manufacturer can access the AAS from the supplier's repository and integrate this AAS into its own systems. Option 2: the manufacturer can transfer the AAS from the supplier's into their own repository, and

[3] https://industrialdigitaltwin.org/content-hub/aasspecifications/specification-of-the-asset-administration-shell-part-2-application-programming-interfaces-idta-number-01002

perform a “transfer of ownership”. We have detailed this transfer of ownership and its value for use cases such as the digital product passport in a dedicated whitepaper.

Software Services: Software services can make use of the published AAS models through the AAS interface to enable visualization, analytics, monitoring, automation, or cross-company collaboration. They are the applications that turn standardized AAS data into actionable value.

3.2 How to realize digitalization use cases with an AAS-based infrastructure

To make this architecture a bit more tangible, let’s see how we can use it to realize the DPP as a bike manufacturer.

Step 1 - Create AAS: We start by creating first AAS models through a model editor. We can start with one of our available bike models as so-called AAS type. Therefore, we can define the Bill of Material (BoM) of this bike type (e.g., which types of wheels, handles, etc. are we using for this type), a digital nameplate. Based on this AAS type, we can create multiple AAS instances, for example bikes which we produce. For each bike, we define a specific component that is used for manufacturing. If the supplier of this component also provides as AAS model containing the product carbon footprint (PCF) data of this component, we can calculate the provided PCF value for each component to retrieve a base value of the PCF of our bike. This value is then increased by additional carbon emissions emitted during manufacturing and transport.

Alternatively, we could also start by modelling multiple example bikes and then generalize these examples into so-called AAS types. Or each example, we can define the components it contains (i.e., the BoM), the nameplate, and the product carbon footprint. Whereas this option introduces redundancy (the same BoM and nameplate must be created for each example), it is often a more natural way to start creating an AAS.

Step 2 - Connect Endpoints: To scale the creation of DPPs to real-world manufacturing, we must connect the AAS models with endpoints and automate AAS model creation through mapping rules. For example, for the DPP of each bike ordered by a customer via our customer portal, we could get the serial number of the ordered bike, the selected bike components and their PCF from the ERP system, and the carbon emissions emitted during the inhouse manufacturing process from the MES. Of course, there are multiple architectural decisions to be considered for this integration (e.g., is data integrated via a publish-subscribe or a client-server pattern, is data pushed into the AAS by the endpoint

or does the AAS proactively get data from the endpoint via respective wrappers), so tool support, domain understanding, and planning is essential for this step. For information regarding components (e.g., the PCF of the saddle which we bought from a supplier), we can get the respective information from the supplier's AAS repository, by either directly accessing the AAS model within their repository or transferring the respective model into our repository through "transfer of ownership".

Step 3 - Develop Software Services: Once we have the AAS models set up, we can start developing software for our use cases and connect this software with our AAS models. As an example in the context of the DPP there is an open-source bike configuration tool available[4], which automatically generates an AAS model of the configured bike, including the PCF value, which is calculated based on

- The PCF of the selected bike components (e.g., saddle, wheel, handlebar) from the supplier's AAS repository, or the manufacturer's repository (after the "transfer of ownership" is performed)
- The PCF omitted by transporting the bike from the manufacturing facility to the customer's address.

From the generated AAS model, the DPP, together with a QR code to be printed on the manufactured bike, can be created.

Generalization to further use cases: Of course, we can realize other use cases with this same architecture. For example, a pre-sales customer portal as a software service can provide access to a digital nameplate, digital handover documentation, or operating instructions for a product and its components. It could even trigger individualized services based on the customer data accumulated in the respective AAS.

Through the AAS interface, the customer portal can always access the most recent data stored in the AAS model. This AAS model is automatically updated, e.g., if contact data provided on the nameplate changes. During manufacturing, the AAS interface can also be used by a predictive maintenance service to regularly check the status of machines and intervene in case of issues. Therefore, the AAS model, connected directly to the physical machine, serves as interoperable data container for information required by the predictive maintenance service.

[4] https://github.com/meta-level-public/aas-bike-showcase

3.3 Feels Overwhelming? Start Small, Scale Later!

The AAS becomes most powerful when it is applied across several use cases. Instead of building isolated solutions, companies can use the same infrastructure to cover very different needs, ranging from operational optimization to service-oriented applications. The advantage lies in scalability: once the foundation is established, additional applications can be added without repeating the setup effort.

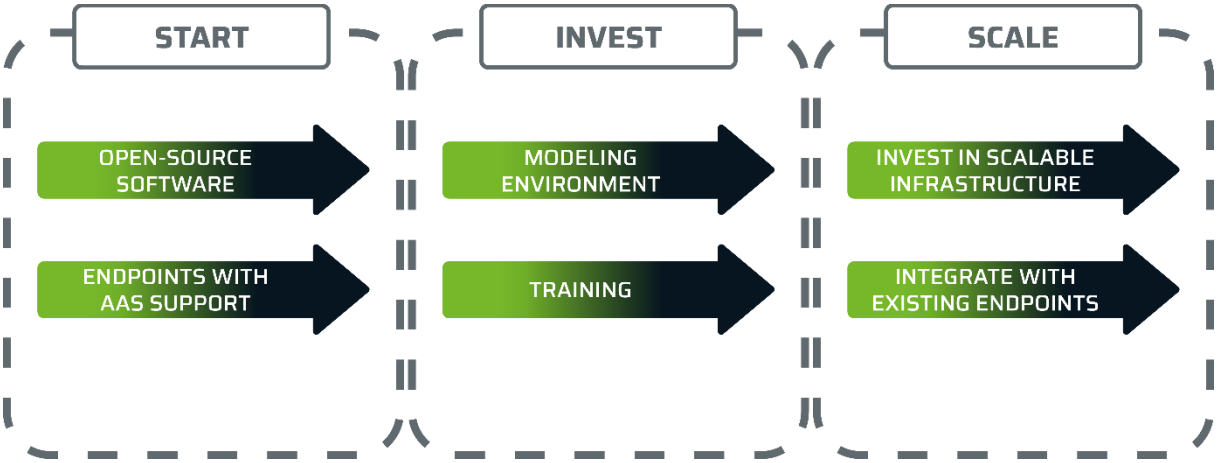


Figure 3: Conceptual process to establish an AAS-based digitalization infrastructure covering multiple use cases

Start. To move from theory to practice, organizations need a way to introduce the AAS step by step. A proven principle is to start small, build experience, and expand over time. The first stage is about exploring what is already available: which data exists, what are quick wins that can be leveraged with this data, and how can first digital twins be created based on this data. The focus should be on creating first tangible results and avoid over-engineered solutions. This can often be done without major investment by using open-source implementations, trial versions or endpoints that already allow export to AAS models. Working with such data gives an immediate entry point and makes the concept tangible or inspires other to try it out themselves.

Invest. The next stage is to make targeted investments that improve efficiency in developing and managing a large number of AAS models, as well as the quality of the resulting models and the overall modelling process. This can include setting up a modelling environment that simplifies the creation, visualization, and maintenance of AAS models, or providing specialized training to internal staff. Depending on the size and commitment of the organization, this may start with individual employees who build expertise and later pass on their knowledge to other departments. Such an approach ensures sustainable competence growth and prepares the organization for larger-scale adoption.

Scale. Finally, scaling requires infrastructure that can handle growth. As the number of models and use cases increases, more robust repositories and interfaces are needed. At this stage, endpoint integration becomes a key factor: AAS models should be automatically created and populated with data from enterprise systems such as MES or ERP, as well as from physical assets. Dedicated endpoint mappers and commercial solutions can provide the scalability and usability needed to support wide adoption across the company and beyond.

In this way, organizations can gradually expand from small experiments to a production-ready AAS infrastructure. Especially for smaller organizations, it is very helpful to include cross-company collaboration in this process, as the value proposition of these companies often depend on components and solutions of their partners. By using a common data structure together with these partners, organizations can act faster, fulfil regulatory requirements, and offer new services, without the need to build their own, complex IT infrastructure.

The following chapter provides an overview of available AAS tools for building such an IT infrastructure and categorizes them to give orientation on what the market already offers.

For readers who want to dig deeper into the strategic aspects of introducing digitalization use cases in an organization, a more detailed roadmap is already available that builds on this overview of available tools and their possible application. While this paper helps to understand which tools can be used for different implementation approaches, the roadmap paper provides step-by-step guidance on how to apply them in practice, along with best practices from real projects. For those interested, the full roadmap can be found in a dedicated paper.

4 Landscape of existing AAS Tools

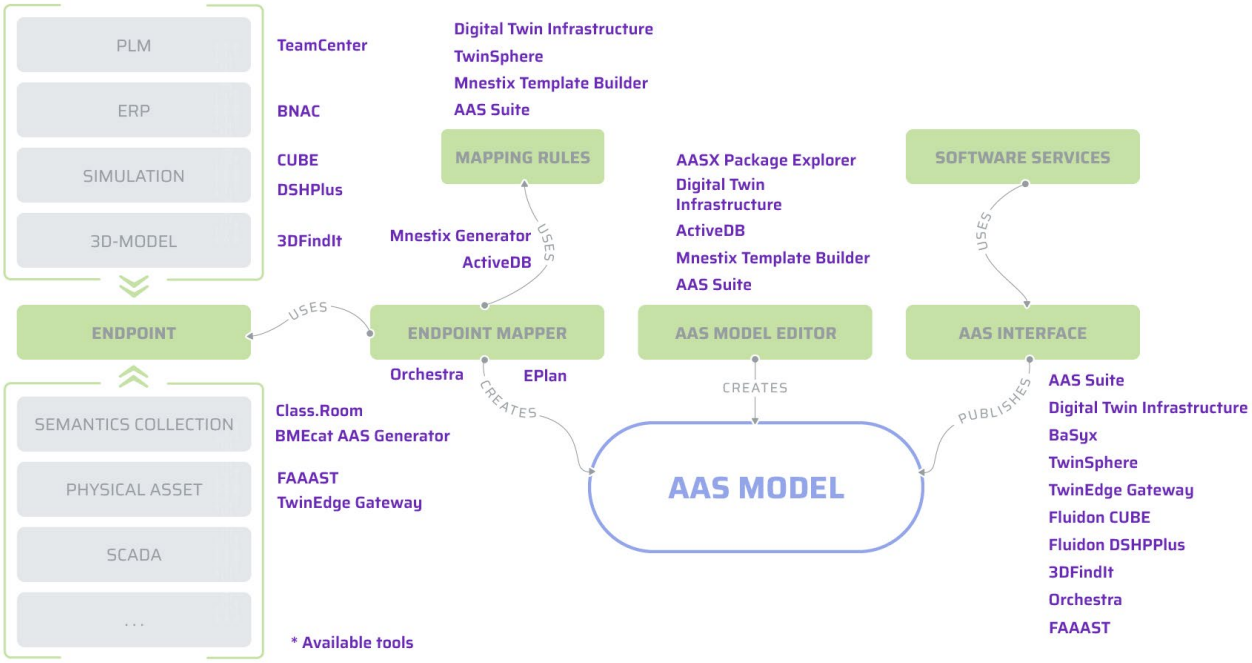


Figure 4: Mapping of identified tools to the components of the architecture presented in Figure 2

4.1 Tool Overview

Here is the table with all listed tools, their characteristics, and a detailed analysis.

TO THE TABLE

From the previously shown architecture, we learned that multiple components need to fit together in order to successfully implement AAS-based digitalization use cases. Typically, industrial tools cover several of these components, although not all of them cover every component. In our tool overview, we therefore aim to identify which components can be realized by which software tools. Accordingly, in the table, we have defined one criterion for each component that indicates whether the respective tool satisfies this criterion.

- **Interfaces:** Can be used as an AAS interface, which serves as the basis for type 2 AAS.
- **Modeling Support:** The tool can be used as an AAS model editor. It is important to differentiate whether the tool allows reading data from existing AAS models or also enables the creation of new AAS models.
- **Endpoint Mapping:** Indicates whether there is support for connecting different endpoints, so that users do not have to start from scratch when automating data

transport into and out of an AAS. If endpoint mapping is supported, the tool can serve as an endpoint mapper and optionally provide a mechanism for defining mapping rules.

After selecting the most fitting toolset for your unique business context, the first step is to set up each tool individually. In the table, we also provide an overview of the channels available to get started with each tool (e.g., whether there is a GitHub repository with a docker-compose file, whether the tool is available as a SaaS solution, or whether individualized configuration needs to be ordered from the tool vendor, see “How to Start” in the table).

4.2 Further Comparison and Insights

The figure in chapter 4 illustrates how the individual tools described in the table fit into the AAS-based digitalization architecture introduced previously. From this overview, together with the details regarding each individual tool as collected in the table, multiple insights emerge.

Insight #1: Different focus points of AAS modelling tools

While the figure shows that many tools offer some form of modelling support, the table reveals that several of them are limited to presenting and visualizing data (depicted as read support in the table). On top of this data presentation, ActiveDB offers the ObjectDesigner to model data structures that can later be exported as .AASX files. However, the focus of the ObjectDesigner is on modelling data that is then made available via ActiveDB. The main purpose of ActiveDB is to provide a scalable, industry-ready solution to access large amounts of data within software. This software can be endpoint mappers or software services. In both cases, ActiveDB generates boilerplate code from the data definitions in the ObjectDesigner (which can include, but are not limited to, imported AAS models) that can be used to interact with the actual data stored in ActiveDB.

The Neoception Digital Twin Infrastructure allows the creation of AAS models defined by templates connected to existing data. Based on these templates, AAS models can be generated and shown by request.

In contrast to these view-only, database-centred, or template-based modelling options, the AAS Designer (part of the AAS Suite) and the AASX Package Explorer are the only modelling tools that allow users to create, view, and edit AAS models with a dedicated focus on the AAS itself. While the AASX Package Explorer is an open-source tool with

certain usability limitations, the AAS Designer offers the full functionality and flexibility of a commercial solution.

Insight #2: Different levels of “data mapping”

Even though integrating data from different endpoints into AAS models is a key focus for all commercial AAS tools, their level of support differs. For some tools, data integration is even the main purpose of their AAS support. FAAAST, TwinEdge Gateway, and SAP BNAC aim to establish the data foundation for AAS files by providing access to the physical asset itself or a Monitoring and Execution System (MES). Other tools expand this scope to specific ecosystems. For example, Fluidon CUBE and DSHPlus enable access to physics-based simulation in connection with AAS models, 3DFindit provides access to 3D models from the existing CADENAS ecosystem, SAP BNAC makes MES data available for use in the AAS, and Siemens TeamCenter offers similar functionality for PLM data.

Orchestra provides access to over 90 ready-to-use adapters for different protocols but does not support extensions. ActiveDB, by contrast, generates boilerplate code to simplify the integration of new tools. Other tools provide mechanisms to define mapping rules that configure an endpoint mapper. These are called feed mappings in the AAS Suite, template rules in the Neoception Digital Twin Infrastructure, mapping rules in the Mnestix Template Builder, and semantic connectors in the TwinSphere Suite.

Insight #3: Extensive open-source tooling available

Existing open-source tools can cover the entire architecture. Many of these tools are collected under the umbrella of the Eclipse Digital Twins project[5]. Interestingly, for several tools from this project, commercial solutions also exist that are built upon the corresponding open-source implementations:

- The BaSyx Enterprise tool by Objective Partner and the AAS Dataspace for Everybody by Fraunhofer IESE both host the open-source BaSyx framework on industry-grade infrastructure as a “software as a service.”
- The Mnestix Generator and Mnestix Template Builder extend the Mnestix Browser; all three projects are maintained by XITASO.

[5] https://projects.eclipse.org/projects/dt

Insight #4: Different levels of AAS support

Except for some open-source tools that focus on visualization and editing of AAS models (e.g., AASX Package Explorer and Mnestix Viewer) and tools that primarily provide data (e.g., SAP BNAC, Siemens Teamcenter), all considered AAS tools offer Type 2 AAS support. Some tools (e.g., ActiveDB and TwinSphere Suite) also explicitly mentioned the scalability of their AAS interfaces to a large amount of AAS models that can be operated simultaneously. Another remark regarding the support of different AAS types is that we did not find any information that would point towards the support of Type 3 AAS for any of the investigated tools.

5 Conclusion

This paper set out to provide orientation in a diverse and sometimes confusing AAS tool landscape. By analysing existing tools and showing how they relate to the underlying AAS architecture, we have demonstrated that a wide range of options already exists to realize digitalization use cases with the AAS today. At the same time, our analysis shows that the ecosystem is still developing and that practical collaboration will remain key to success.

The AAS has proven its role as the central element for achieving interoperability in industrial digitalization. It creates a common language that connects engineering, production, and operation, and enables data to flow seamlessly between tools, systems, and organizations. Especially for smaller organizations that don't have the power to build their own, complex IT infrastructure, this cross-company interoperability is a prerequisite for materializing the potentials of digitalization. The examples and tool mappings in this paper show that the foundations for this vision are no longer theoretical. With the right setup, it is possible to start building real value today, from digital product passports to condition monitoring or automated handover documentation.

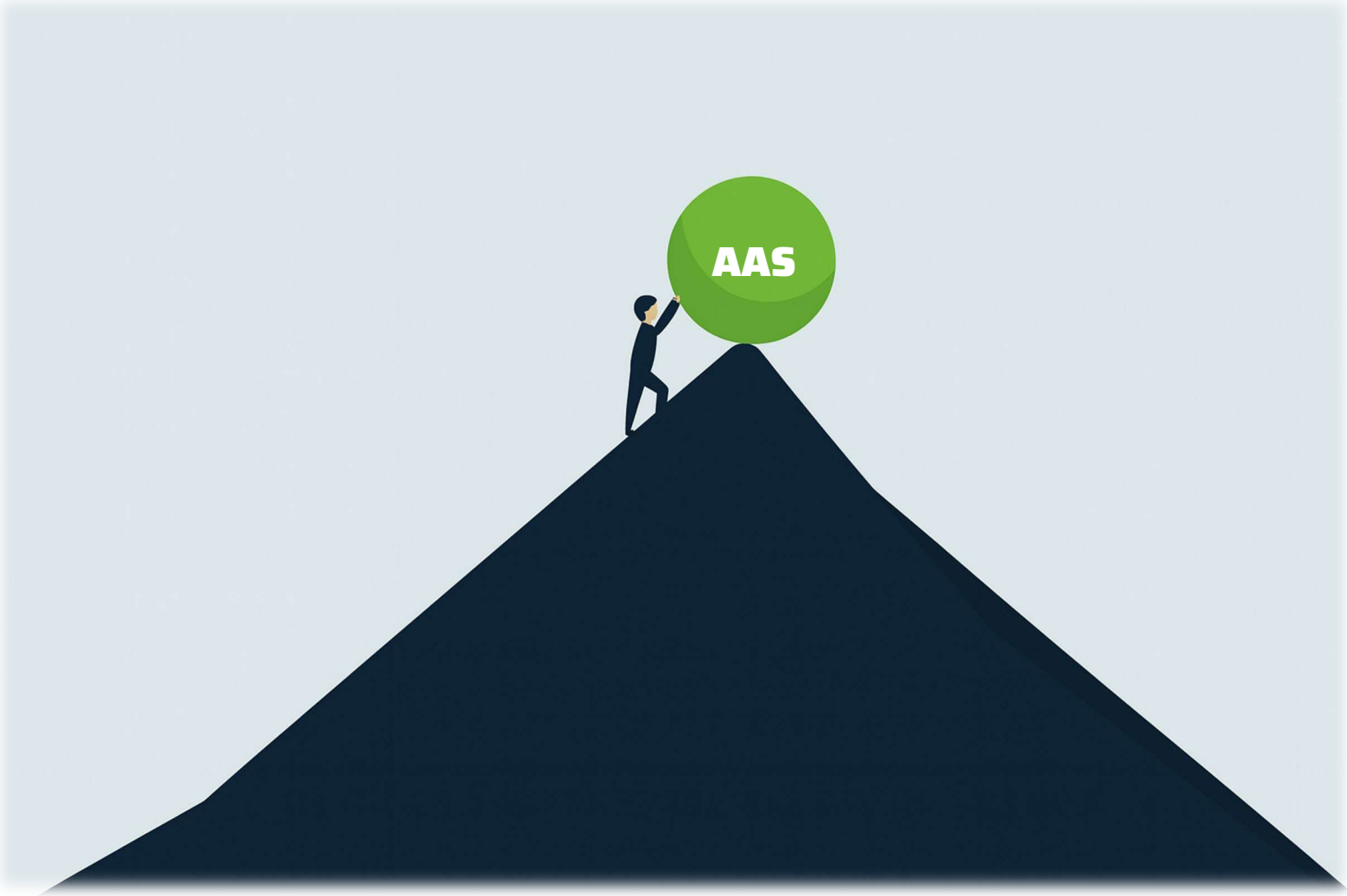
Before selecting or integrating tools, organizations should define a clear digital roadmap that reflects their goals, maturity, and available data sources. This roadmap describes the preconditions for success: ownership for AAS topics, clarity on data structures and governance, and at least one concrete use case that can deliver measurable benefit. The corresponding paper can be found here.

Although the available tools cover the full architecture, not every component is equally mature. In particular, improved support for standardized mappings, automated instance generation, and plug-and-play interoperability between tools is still needed. Often, a combination of multiple tools is needed. Additionally, experiences from the "Follow the Twin Challenge" performed by the Open Industry 4.0 Alliance showed that interoperability between different AAS tools still incurs some challenges when it comes to the details of actually connecting AAS models between tools of different vendors. Users play a key role in driving the progress of pushing tool providers towards a common, interoperable AAS tool landscape: demand these capabilities from your tool providers, share feedback from your implementations, and join common industry activities such as hackathons or working groups. Only through this dialogue, the ecosystem can converge towards practical, interoperable solutions.

In theory, interoperability is ensured by the AAS standard itself. In practice, however, the many hackathons and joint demonstrators within the Open Industry 4.0 Alliance have shown that the real challenges often lie in the details, such as differing interpretations of

specifications, missing mappings, or incompatible interfaces. The best way to uncover and solve these issues is to collaborate, test tools in mixed environments, join hackathons, and verify interoperability hands-on.

The AAS has reached a level of maturity where industrial adoption can scale. What comes next is less about waiting for perfect tools and more about taking the next step. The time to act is now: start small, test broadly, and cooperate openly. Each contribution drives the topic forward and supports the AAS in becoming a practical, widely adopted standard that drives digital transformation in manufacturing industry.



TALKERS TALK

WE IMPLEMENT

Revision History

Date	Rev	Chapter	Description
2025-11-24	1.0.0	-	First Version
2025-12-17	1.1.0	3.1 & 4	Illustrations; added further tools, changed “MES” to “SEMANTIC CONNECTIONS”