

AN AVIATION DATASPACE ARCHITECTURE FOR AUTONOMOUS AND SOVEREIGN MRO AND MANUFACTURING COLLABORATION WITH FEDERATED SERVICE NEGOTIATION

M. Weiss^{*)}, F. Detig⁺⁺⁾, M. Eckert⁺⁺⁾, F. Raddatz⁺⁾, G. Wende⁺⁾

^{+) DLR, Institute for Maintenance, Repair and Overhaul, Hamburg, Germany}

^{++) DLR, Institute for AI Safety and Security, Sankt Augustin, Germany}

Abstract

Cross-organizational collaboration is becoming increasingly important in aviation MRO and manufacturing, for example to diagnose anomalies, coordinate service actions and exchange auditable evidence. However, existing approaches tend to focus on either semantic asset representation or governed data exchange. The integration of these two areas for autonomous, service-oriented collaboration in aviation remains insufficiently addressed. This paper presents a dataspace architecture that combines proactive Asset Administration Shells with sovereign dataspace mechanisms, focusing on the MRO domain. This architecture enables assets and business applications to act as smart digital representatives that can describe their capabilities and skills, derive service needs from a Bill of Process, discover suitable providers, negotiate service interactions and invoke services via local or dataspace-governed communication paths. In this architecture, AASs provide the semantic and agentic layers for asset- and service-oriented interactions, while EDC-based mechanisms support contractual access control, policy enforcement and controlled evidence exchange. This concept is demonstrated through an aviation MRO scenario involving requester assets and distributed smart service providers.

1. INTRODUCTION

1.1. Motivation: Distributed Aviation MRO and Manufacturing Networks

The economic relevance is substantial: global commercial aviation MRO demand reached approximately USD 136 billion in 2025 and is forecast to approach USD 193 billion by 2036, while the worldwide commercial aircraft fleet is expected to grow from about 30,000 aircraft in 2026 to more than 41,000 in 2036 [30]. Meeting this growing demand efficiently requires aviation MRO and manufacturing activities to be increasingly organized as distributed value networks. Aircraft operators, MRO providers, component manufacturers, inspection service providers, production resources and digital service providers must collaborate across organizational boundaries to diagnose anomalies, exchange evidence, verify conformity, trigger follow-up services and maintain lifecycle traceability. A representative blueprint for this challenge is connected end-to-end quality management in aerospace supply networks, as investigated in previous Aerospace-X work [11]. Figure 1 summarizes the corresponding target architecture: process monitoring data, inspection results, certification evidence, non-quality information and lifecycle documentation are represented through semantically standardized digital asset structures and exchanged through governed dataspace mechanisms. The figure therefore does not introduce a separate quality-management use case, but illustrates the general architectural problem addressed in this paper: aviation stakeholders require traceable, interoperable and sovereign interaction across organizational, system and lifecycle boundaries.

This motivation is embedded in current aviation-oriented dataspace initiatives. Aerospace-X [1] transfers Manufacturing-X principles to aerospace supply networks by

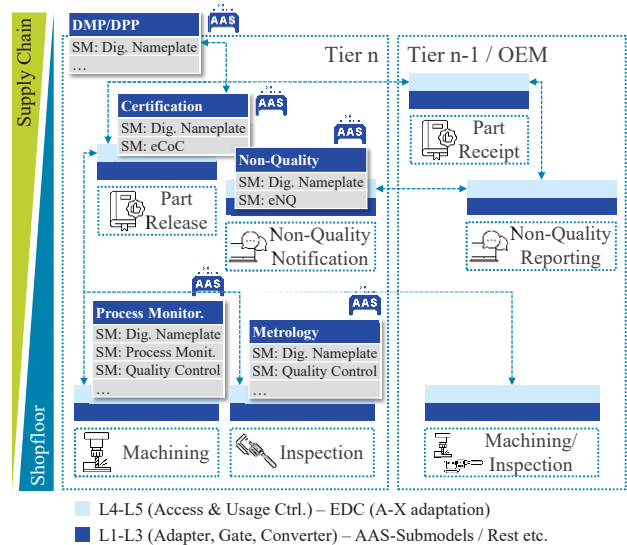


Figure 1: Target architecture for connected end-to-end quality management in aerospace supply networks via dataspaces [11]

developing a collaborative ecosystem for digital continuity, data sovereignty and data-based value creation. DECADE-X continues this direction as an operative sector-specific dataspace ecosystem for aerospace and defense [3]. Both initiatives show that future aviation ecosystems require architectures that enable collaboration without centralizing raw data or giving up control over data, interfaces and service conditions. In this context, the MX-Port concept provides the architectural reference structure. Its layers for adapter, converter, gate, access and usage control, and discovery describe how heterogeneous data sources and applications can be connected to interoperable, policy-controlled dataspace interaction [8].

^{*) Corresponding author: Marco.Weiss@DLR.de}

Earlier publications have addressed important parts of this challenge. As discussed in [27], [28], Industry 4.0 and cyber-physical production systems provide the foundation for interoperable and modular industrial systems, including smart Prognostics and Health Management (PHM) systems. In aviation MRO, predictive maintenance, Life Cycle Engineering and Digital Product Passport concepts require consistent data acquisition, semantic traceability and stakeholder-specific access to lifecycle information [39]. Our previous work applied proactive Asset Administration Shells (AAS) for autonomous event management, Digital Product Passport exchange, quality inspection and condition-driven behavior in aviation MRO and manufacturing [37], [38], [39]. The present paper builds on these results but changes the focus from individual quality, inspection or condition-monitoring scenarios toward the architecture and techniques required for an aviation dataspace. Within this architecture, the AAS is positioned as the semantic layer, data bridge and smart digital representation of real-world assets and services, while EDC-based dataspace mechanisms provide contractual access control, policy enforcement and controlled evidence exchange.

1.2. Problem Statement: Dataspace Architecture, Semantic Interoperability and Service Interaction

Three challenges must be solved jointly: First, cross-organizational aviation MRO and manufacturing require a dataspace architecture that enables distributed stakeholders to exchange data, evidence and service access without centralizing raw data or giving up control over interfaces, policies and usage conditions. Second, the dataspace requires a semantic bridge to real-world assets. Dataspace technologies can govern access to data assets and endpoints, but they do not define the domain semantics of aircraft components, inspection results, maintenance evidence or service capabilities. The Asset Administration Shell (AAS) addresses this gap by providing a standardized digital representation of assets, submodels, identifiers, operations and interfaces [5], [13], [15]. Third, aviation MRO and manufacturing require active service interaction, not only passive data access. Asset-related events may create needs for diagnosis, inspection, repair, documentation or evidence exchange. A proactive AAS can detect relevant states, formulate service needs, identify providers, negotiate interaction conditions and trigger services. Local or trusted interactions may still use conventional AAS, REST, OPC UA or message-based communication, while cross-company interactions with sovereignty requirements can be routed through dataspace mechanisms.

The resulting research gap lies in the architectural integration of these perspectives. Existing AAS-based approaches provide semantic interoperability and asset representation. Dataspace approaches provide governance, contracts and controlled access. What is still needed is an aviation dataspace architecture that explains how AAS-based assets and services can be combined with dataspace principles to support sovereign service interaction.

1.3. Research Objective and Questions

The objective of this paper is to describe and demonstrate an architecture for flexible, sovereign and autonomous service interaction in aviation MRO and manufacturing. The starting point is the need for flexible MRO production systems that can react to asset states, anomalies, inspection needs and documentation requirements across

organizational boundaries. This need is addressed through an MRO-as-a-Service principle in which physical, digital and hybrid service functions are described as capabilities and realized through provider-specific skills. A dataspace architecture provides the governed interaction environment for cross-organizational access, while proactive AAS act as smart digital representatives of assets and services. They provide the semantic layer, bridge heterogeneous applications and initiate, negotiate and document service interactions. This leads to the following research questions:

- RQ1: How can aviation MRO and manufacturing collaboration be described as a flexible MRO-as-a-Service ecosystem based on service requesters, service providers, capabilities, skills and BoP-driven service composition?
- RQ2: How can a dataspace architecture based on Manufacturing-X and MX-Port principles provide the governance mechanisms required for sovereign cross-organizational service interaction in this ecosystem?
- RQ3: How can proactive AAS act as smart digital representatives of aviation assets and services by providing semantic modelling, data bridging, capability-based negotiation and AAS-native or EDC-governed service invocation?

1.4. Contribution and Paper Structure

The contribution of this paper is an architecture-oriented extension of previous work towards a federated aviation service ecosystem. The paper follows the logic: first, aviation MRO and manufacturing require flexible production and service systems; second, this flexibility is described through the MRO-as-a-Service principle; third, cross-organizational interaction is governed through a dataspace architecture; and fourth, proactive AAS provide the semantic and agentic mechanism that makes this architecture operational.

The paper is structured as follows. Chapter 2 introduces MRO-as-a-Service in aviation and describes service requester and provider roles, capability- and skill-based abstraction, BoP-driven service composition and choreographic service negotiation. Chapter 3 introduces the dataspace concept and maps the aviation scenario to the MX-Port layers, including the distinction between EDC catalogues, federated catalogues and AAS registries. Chapter 4 explains how proactive AAS represent assets and services, bridge heterogeneous applications and enable capability-based interaction through AAS-native or EDC-governed communication paths. Chapter 5 demonstrates the resulting architecture in the aviation dataspace demonstrator using selected business applications as AAS-described smart service providers.

2. MRO-AS-A-SERVICE IN AVIATION

This chapter introduces MRO-as-a-Service as the functional service model used in this paper. MRO-as-a-Service does not mean that maintenance is moved to a cloud platform. It means that physical, digital and hybrid MRO functions are described as modular, discoverable and negotiable contributions. The requester does not primarily search for a specific application or organization, but for a required capability. Providers offer skills that realize this capability under defined technical, organizational and access conditions.

The approach and terminology follow the Capability-Skill-Service (CSS) model [4]: A capability describes an implementation-independent function or effect. A skill is the

executable provider-specific realization of such a capability. A service describes the provision of one or more capabilities by a service provider to a service requester under defined service properties, such as availability, quality, documentation obligations, access conditions or contractual constraints. Technical APIs, file interfaces or communication endpoints are therefore not services in this strict sense, but access mechanisms used to invoke or support a skill.

2.1. Capability-Based Service Roles in MRO

The conceptual starting point is the separation of process requirements from concrete resources. In classical MRO environments, a process step is often assigned to a known workshop, system, application or stakeholder. This is efficient for established routines, but limits flexibility when the asset state changes, when a provider is unavailable, when additional evidence is required or when an external provider must be involved. A capability-based view introduces a functional abstraction between the required process effect and the resource that performs it.

In this paper, a service requester is an entity that requires a specific MRO-related effect under defined conditions. A service provider is an entity that offers one or more skills to realize such an effect. For the requester, the relevant information is not the internal implementation of the provider, but the offered capability, the required inputs, the expected outputs, the evidence obligations and the applicable service conditions. A provider proposal may therefore include the offered capability, the selected skill implementation, execution parameters, expected result quality, validity period and organizational or contractual constraints.

The aviation MRO ecosystem comprises functional service roles along the lifecycle of aircraft, systems, components and parts. These roles include maintenance, disassembly, inspection, repair, overhaul, assembly, testing, decision support and evidence management. They may be provided

or performed by certified MRO organizations, Original Equipment Manufacturer (OEM) units, component workshops, inspection providers, test facilities, engineering departments, digital service providers or combinations of equipment, software and human expertise.

Table 1 summarizes representative MRO service roles from a capability-based perspective. For each service function, it lists the operational purpose, example capabilities and typical stakeholder groups. The table is not a normative taxonomy, but a domain abstraction used to show how MRO functions can be decomposed into reusable capabilities and later combined into service-specific process chains. The table also illustrates how the same abstraction can be applied to different MRO functions. For instance, the inspection row describes the service function "Inspection" as the determination of condition, conformity or damage state. This function may be decomposed into capabilities such as object scanning, CAD processing and conformity evaluation, which can be provided by inspection providers, Non-Destructive Testing (NDT) providers, quality departments, MRO providers or OEM units. In contrast, the repair row describes a different service function, namely the correction of a detected defect or non-conformity. Its example capabilities include repair assessment, repair execution and repair verification. Thus, the service function defines the operational context, while the capabilities describe reusable functional contributions that may later be selected and combined in a service-specific process chain.

Service needs can originate at different system levels. An aircraft or unmanned aircraft system may require condition assessment at system level. An engine may require sub-system-level inspection. A component or part, such as a compressor blade, may require geometric analysis, repair assessment or evidence documentation. MRO-as-a-Service therefore covers both isolated service requests and composed service processes across physical, digital and hybrid providers.

MRO-Service	Description	Example Capabilities	Concerned Stakeholders
Maintenance	Maintain an asset according to scheduled, unscheduled or condition-based requirements	- Maintenance planning - Task execution - Documentation	MRO provider, aircraft operator, CAMO, OEM service unit
Disassembly	Remove or disassemble an item from a higher-level system, subsystem or assembly	- Disassembly planning - Disassembly execution - Part identification	MRO shop, engine shop, component workshop, OEM repair unit, specialized tooling provider
Inspection	Inspect a part, component, subsystem or system to determine condition, conformity or damage state	- Object scan - CAD processing - Conformity evaluation	Inspection provider, NDT provider, quality department, MRO provider, OEM, certifying staff
Repair	Correct a detected defect, damage or non-conformity to restore an item to an acceptable condition	- Repair planning - Repair execution - Repair verification	Repair station, MRO repair shop, OEM service unit, component supplier, engineering support unit
Overhaul	Restore the serviceability of a system according to defined interval, work scope or lifecycle requirement	- Work-scope definition - Module overhaul - Release preparation	Certified overhaul provider, engine MRO, component MRO, OEM, specialized repair center
Assembly	Assemble, install or reinstall an item into a higher-level system or assembly	- Assembly planning - Assembly execution - Configuration verification	MRO provider, production unit, engine shop, component workshop, OEM, airline maintenance
Assessment	Support maintenance, repair, routing or follow-up decisions based on asset state, evidence, constraints or process context	- Condition assessment - Cost-benefit assessment - Risk evaluation - Advisory	CAMO, airline engineering, MRO engineering, OEM engineering, analytics provider, digital service provider
Testing	Verify function, serviceability, plausibility or conformity of a result	- Functional test - System check - Acceptance evaluation	Test bench provider, simulation provider, MRO engineering, OEM, quality assurance, certification unit
Persistence	Persist, transform, standardize and provide digital artefacts, metadata and evidence	- Storage - Translator	Storage provider, digital service provider, airline, MRO provider, OEM, quality department, CAMO

Table 1: Example MRO services, capabilities and potential stakeholders

2.2. BoP-Driven Composition and Choreographic Service Negotiation

The Bill of Process provides the process logic for combining capabilities. It defines which effects must be achieved, which capabilities are required, in which order or under which conditions they should be executed, and which intermediate results or evidence records are expected. The BoP does not prescribe a fixed software chain or service granularity. Depending on its decomposition, the requester may tender individual capabilities or an aggregate service whose provider organizes the subordinate steps.

A repair-related BoP may include disassembly, inspection, repair execution, validation, documentation and assembly. An inspection-related BoP may include artefact retrieval, scan acquisition, data conversion, deviation analysis and evidence storage. A condition-related BoP may include data access, condition assessment, decision support and documentation. At the selected granularity, the requester derives the next required elementary or aggregate capability from the BoP and searches for suitable providers.

The interaction follows a choreographic model. The requester derives the required capability and sends a Call for Proposal (CfP) to a registry-integrated message broker, which identifies registered and active providers and distributes the request. Potential providers evaluate the request against their offered capabilities, skill implementations, parameters, availability, resource state and service conditions. This evaluation corresponds to a feasibility check: the provider assesses whether the requested effect can be achieved in the concrete execution context. If the feasibility check is positive, the provider returns a service offer. The requester compares the offers and selects the provider whose proposal best satisfies the required capability, the BoP constraints and the applicable governance requirements. The selected provider then executes the corresponding skill and returns the result, artefact or evidence reference required for the next BoP step.

This choreography describes service negotiation at the functional level. It determines whether a provider can realize a requested capability through a suitable skill and under which service conditions. It does not yet determine the technical or contractual access path to the provider endpoint. In cross-organizational settings, this access path is governed separately by the dataspace mechanisms introduced in Chapter 3. Chapter 4 then explains how the proposed architecture represents and executes this model technically.

3. DATASPACE CONCEPT

Chapter 2 described aviation MRO-as-a-Service as a flexible service ecosystem in which requester and provider roles interact through capabilities, skills, BoP logic and functional service negotiation. This chapter introduces the dataspace concepts required to govern such interaction across organizational boundaries. The focus is not on explaining all dataspace technologies in detail, but on identifying the mechanisms that are relevant for the proposed architecture: participants and connectors, catalogues and discovery, dataspace contract negotiation, controlled access, and the MX-Port layering.

A dataspace is needed because aviation MRO services often involve safety-relevant, certification-relevant, proprietary or commercially sensitive data. A requester may need

to invoke a service provided by another stakeholder, but the provider must remain in control of endpoint visibility, access conditions, usage policies and contractual constraints. The dataspace therefore provides the sovereign interaction environment, while the AAS-based service model provides the semantic and operational logic of the requested service.

The chapter first summarizes the dataspace principles and relevant X-initiatives. It then clarifies the technical interaction model, with particular attention to the distinction between an EDC catalogue, a federated catalogue and an AAS registry. Finally, it maps the proposed aviation architecture to the MX-Port concept and motivates the Hercules-dominant hybrid configuration used in this paper.

3.1. Dataspace Principles and X-Initiatives

A dataspace is a decentralized ecosystem in which participants exchange data products, services or endpoint access under commonly agreed rules [29]. Instead of moving all data into a central repository, providers expose controlled offerings through connectors. Consumers discover these offerings, negotiate access conditions and use the agreed access path. This principle supports data sovereignty because providers can define the terms under which their data, interfaces or services are made available.

Relevant X-initiatives provide the industrial context for this work. Catena-X [31], [2] demonstrates a collaborative automotive data ecosystem, while Tractus-X provides open-source components used in Catena-X and Manufacturing-X environments. Manufacturing-X extends the approach toward cross-sector industrial data ecosystems, while Factory-X [9] operationalizes it through the MX-Port concept. For aviation, Aerospace-X and DECADE-X transfer these principles into aerospace-oriented ecosystems. The present paper therefore uses Manufacturing-X and MX-Port as an architectural basis, but adapts them to MRO-specific requirements such as lifecycle traceability, functional service negotiation and maintenance evidence.

3.2. Protocols and Technical Components

A connector represents a participant in the dataspace and controls access to data sources, services, runtimes and endpoints provided as assets. To ensure data sovereignty, a connector's architecture is logically split into a control plane for governance and a data plane for data exchange. The Dataspace Protocol (DSP) defines the interoperable control plane interaction between these connectors. It specifies how catalogues are retrieved, usage policies are expressed, contract agreements are negotiated and transfer processes are initiated.

In the Tractus-X-based EDC implementation used in this work, this functionality is physically separated into control plane and data plane server processes: The control plane manages resources, policies and contract definitions, performs catalogue and contract interactions and coordinates transfer-process initiation. The data plane executes the governed transfer or provides controlled access to the underlying data source or service endpoint. This separation decouples policy- and contract-intensive control operations from the actual transfer of data streams [7]. The connector deployment is complemented by an Identity Hub, which manages decentralized participant identities and verifiable credentials required for trusted dataspace interaction [6], and by a Vault, which protects cryptographic keys, secrets and connector credentials [14]. These components

establish the identity, governance and transfer infrastructure of the dataspace; they do not determine the domain-specific purpose or engineering semantics of an interaction.

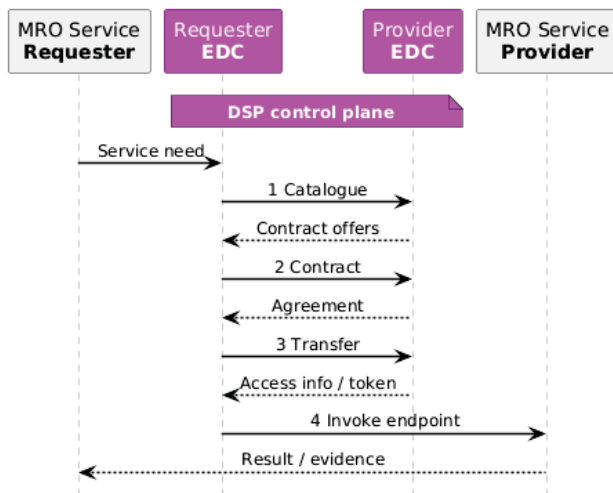


Figure 2: Dataspace Protocol Sequence Diagram

A central element of this interaction is the catalogue. A dataspace catalogue is a machine-readable description of the offerings that a provider makes available to potential consumers. These offerings are not necessarily the raw data itself, but metadata descriptions of datasets, service endpoints or access points combined with policies defining the conditions under which access can be negotiated. Catalogue retrieval is therefore the first step by which a consumer identifies governed resources offered by another participant.

During catalogue retrieval, the consumer receives the provider's negotiable contract offers, usually filtered according to policy and participant context. After a suitable offer has been selected, the consumer initiates contract negotiation. A successful negotiation creates a reusable contract agreement, while the subsequent transfer process provides the access information required to use the governed resource, for example an endpoint data reference and an associated access token. Against this background, this paper distinguishes three catalogue or registry concepts:

1. The EDC provider catalogue describes negotiable dataspace offerings of one provider connector. In the proposed architecture, such offerings may represent governed AAS access points, for example an "/i40commu" path endpoint, an AAS REST endpoint, an AAS registry endpoint or a service API.
2. A federated catalogue supports discovery across multiple providers or connector catalogues. It reduces the need for consumers to know every provider connector in advance and can act as a dataspace-wide discovery layer.
3. The AAS registry is not a dataspace catalogue in the contractual sense. It stores AAS descriptors, asset identifiers, submodel endpoints and AAS-related service information. In the demonstrator, it is integrated with a message broker that uses the registry state to distribute CfPs. The registry identifies which AAS represents an asset or service and where it is reachable, whereas the EDC catalogue identifies which governed offering can be negotiated.

This distinction is important for the proposed MRO service architecture. The AAS registry and its integrated broker

support semantic discovery and CfP distribution, while the EDC catalogue supports contractual discovery and access negotiation. Alternatively, a requester could retrieve authorized provider descriptors and address providers itself; the demonstrator uses broker-mediated distribution to avoid exposing the complete provider endpoint set. The AAS registry identifies the digital representative and its semantic endpoints, whereas the EDC catalogue governs access to selected endpoints as dataspace offerings.

3.3. MX-Port-Based Architecture

The MX-Port concept [8] provides the architectural structure used to transfer Manufacturing-X principles into the proposed aviation MRO and manufacturing scenario. It defines five layers: L1 Adapter, L2 Converter, L3 Gate, L4 Access & Usage Control, and L5 Discovery. The MX-Port document states that the concept structures the complete technical data exchange stack from the data source to exchange within or between companies, and that not all layers need to be used in every application.

For the present paper, the MX-Port model is important because it prevents the dataspace architecture from becoming a monolithic connector-only view. It clarifies that an aviation dataspace must integrate heterogeneous asset and application interfaces, provide semantic conversion, expose harmonized interfaces, enforce access and usage control, and support discovery. This directly supports RQ2: an MX-Port-based dataspace architecture can be transferred to aviation MRO and manufacturing by mapping aviation assets, service endpoints, AAS submodels, EDC connectors, and registries to the five MX-Port layers.

Factory-X defines several MX-Port configurations. The Hercules configuration uses DSP/DCP for the upper discovery and access/usage-control layers and AAS REST/AAS submodels for the gate and converter layers. The Leo configuration uses ID-Link and AAS Security at the upper layers, while also relying on AAS REST and AAS submodels. The Orion configuration focuses on OPC UA-based shop-floor integration while sharing the upper dataspace-oriented layers with Hercules. Current Factory-X material confirms that Hercules, Leo, and Orion are preselected MX-Port configurations under development or realization.

The architecture proposed in this paper is best described as a Hercules-dominant hybrid. It is Hercules-dominant because cross-company endpoint access is governed by EDC/DSP/DCP mechanisms. However, it remains compatible with Leo because the AAS itself may enforce fine-grained access to submodels, properties, operations, and routes through AAS Security. This is essential for aviation MRO: access to an endpoint does not automatically imply access to every maintenance record, condition indicator, DPP element, or inspection result. AAS Part 4 [16] explicitly frames dataspace interoperability with AAS as requiring common models, APIs, and security mechanisms, including identity technologies, identity providers, token claims, AAS security, registries, and discovery mechanisms.

This leads to a layered interpretation of the proposed architecture. At L1, adapters connect physical assets, inspection systems, service applications, and back-end systems. At L2, AAS submodels provide the semantic model for lifecycle, condition, quality, DPP, capability, and service information. At L3, AAS endpoints and service APIs act as gates. At L4, EDC contracts and policies govern cross-company access, while AAS Security can later refine access

inside the AAS. At L5, discovery combines dataspace catalogues, participant discovery, AAS registries, and service discovery.

4. PROACTIVE ASSET ADMINISTRATION SHELL

Chapter 2 defined the capability-based service model, while Chapter 3 introduced the dataspace environment. This chapter explains how the proactive AAS operationalizes their integration. It provides the semantic representation of assets and services, bridges heterogeneous operational systems and executes event-driven service interaction. The implementation builds on the server-based proactive AAS framework developed in previous DLR work [38], [40], [41] and on related research concerning proactive and extensible AAS implementations [12], [25], [26], [32].

4.1. AAS as Semantic Asset Layer

The AAS is the semantic representation of aviation assets [42]. Aircraft components, inspection systems, robotic stations, service applications, and registries are represented as Industry 4.0 Components by coupling the real or logical asset with its AAS. The AAS provides the standardized digital structure in which asset identity, lifecycle data, condition information, quality evidence, service descriptions, and Digital Product Passport elements can be represented in a machine-interpretable form. This semantic layer is required because aviation MRO and manufacturing processes do not only exchange data values, but they also exchange engineering meaning. A deviation report, a remaining-useful-life estimate, a non-conformity, a repair request, or an inspection result is only useful if it remains connected to the asset, the lifecycle phase, the involved service, the measurement context, and the evidence chain. The AAS provides this connection by organizing information in submodels and submodel elements. It therefore acts as the semantic anchor of the MRO service ecosystem.

In this architecture, relevant AAS submodels include nameplate and technical data, documents, condition records, time-series data, predictive-maintenance indicators, Digital Product Passport structures [24], quality-control information [23], service request information, capability descriptions [21], skill descriptions [19], and execution results. These submodels make it possible to describe both static asset information and dynamic service-relevant information. For example, an inspection task can refer to geometry data, measurement requirements, surface features, tolerance thresholds, and expected result formats. A repair task can refer to detected events, required capabilities, tools, constraints, and confirmation records.

4.2. AAS Runtime and Data Bridging

The proactive AAS bridges heterogeneous sensors, controllers, inspection devices, applications, repositories and service back ends with the semantic AAS model. Operational data are acquired and preprocessed by an Asset Data Server and exposed through interfaces such as REST, OPC UA, MQTT or file-based mechanisms. The AAS server maps these data to submodels and provides the corresponding northbound interfaces, operations and communication endpoints.

In the DLR implementation, these server components are deployed as an Industry 4.0 Component Stack (I4.0CS), as

described in previous work [38], [40], [41]. The I4.0CS combines the represented physical or logical asset, an Asset Data Server and the proactive AAS server. For dataspace participation, it additionally hosts participant-local EDC control plane and data plane servers, an Identity Hub and a Vault. The I4.0CS is therefore an implementation environment and not an additional element of the normative AAS metamodel.

The proactive AAS server maintains the information model at runtime and integrates data management, communication adapters, an internal message and event bus, decision algorithms and finite state machines. These mechanisms support condition monitoring, event interpretation, BoP processing, capability and skill evaluation and requester-provider interaction.

4.3. AAS-Based Capability and Skill Modelling

The AAS provides the modelling framework for the capability and skill abstraction introduced in Chapter 2. As shown in Figure 3, a requester AAS can describe a required capability by referencing a semantically unique capability identifier and by specifying the required properties, inputs, outputs and constraints. A provider AAS can advertise provided capabilities and associate them with executable skill implementations. This semantic layer is based on the IDTA Capability Description submodel [21].

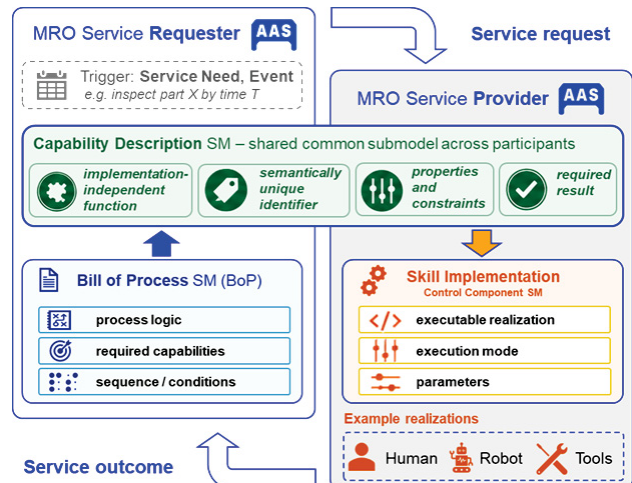


Figure 3: Semantic interoperability through a shared Capability Description submodel across requester and provider

On the provider side, a capability-related skill describes how the required function is realized by one or more skill implementations. This includes execution modes, parameters, input and output requirements, constraints, quality indicators and service-specific properties. In the proposed architecture, such skill information can be represented by using standardized AAS structures such as the IDTA Control Component descriptions [18], [19]. A service request can additionally be represented using service-request-oriented AAS structures, while interface details can be described using asset interface descriptions [20], [22]. This allows the requester to perform application-agnostic matching. The requester AAS does not need to know whether a provider internally uses a web application, AI model, simulation tool, database, robot, inspection device or manual decision-support process. It evaluates the provider's advertised capability, the associated skill, the expected result structure and the applicable access conditions.

4.4. Proactive AAS as Service Agent

The third role of the AAS is its proactive behavior. In contrast to a passive digital representation that only exposes information on request, a proactive AAS acts on behalf of its represented asset. It evaluates asset states and events, derives service needs, identifies suitable providers, negotiates service provision, invokes the selected skill and processes the resulting evidence [12], [38], [40].

In the DLR implementation, this behaviour is realized by interacting finite state machines and decision modules. An Event Manager evaluates condition indicators and maps relevant events to service needs. A Production Manager interprets the task-specific BoP and derives the next required elementary or aggregate capability. Service Requester and Service Provider modules implement the Call-for-Proposal interaction by creating, evaluating and responding to capability-based requests.

As shown in Figure 3 (top left), a service interaction begins with an event, which may be triggered by a condition threshold, an inspection requirement, missing evidence, an operational anomaly, or a production step. The AAS service requester then evaluates the event in terms of the task-specific BoP (see Figure 3, bottom left), deriving the next required capability and its related properties, along with their runtime values. A service request notification [17] including the CfP is then created and sent to potential provider AASs via the message broker integrated with the AAS registry. The AAS service providers then evaluate the request against their available capabilities, skill implementations, execution parameters, stakeholder roles and access conditions (see Figure 3 on the right). If the service is feasible, they return proposals. The requester AAS then evaluates the proposals received, selects the most suitable provider and accepts the offer by ordering the task. It will later receive confirmation and feedback on the execution of the task, as well as evidence that the service request has been completed.

4.5. AAS-EDC Service Integration

The proactive AAS can operate in local, trusted or pre-integrated environments without dataspace infrastructure. In this case, requester, registry and provider AAS instances exchange service-related messages through AAS-native communication paths, such as AAS interfaces, REST, OPC UA, MQTT or file-based mechanisms. For cross-organizational interaction, the same semantic messages can be transmitted through EDC-governed endpoint access, policy enforcement and auditable transfer.

The proposed architecture implements a dataspace-aware dispatch mechanism in which the AAS acts as the application-level orchestration authority for the participant-local EDC control plane and data plane. Based on the asset state, BoP and current interaction context, the AAS determines the purpose of the communication, the required counterpart and the endpoint to be accessed. The EDC does not derive this domain-specific intent; it performs the requested dataspace operations while independently enforcing the provider's identity, policy and contract conditions.

Before transmitting a service-related message, the AAS resolves the target AAS and the corresponding EDC asset. It first checks whether a valid contract agreement and usable

endpoint data reference are already available. Existing agreements and transfer information can be reused to avoid unnecessary control plane interactions. If no valid route exists, the AAS invokes the management interface of its local EDC to retrieve the remote catalogue, identify the contract offer associated with the required endpoint, initiate contract negotiation and start the transfer process. The Identity Hub provides the participant identity and verifiable credentials required for this interaction, while the Vault protects the associated keys, secrets and connector credentials. After successful transfer initiation, the EDC supplies an endpoint data reference and access token through which the data plane exposes the governed endpoint. The AAS then transmits the semantic service message through this route.

This mechanism is applied independently by each active participant in the service choreography. As illustrated in Figure 4, after an event, the requester AAS derives the required capability from the BoP and sends a Call for Proposal (CfP) to the message-broker function of the Registry AAS. For governed communication, this requires a previously negotiated dataspace contract with the registry/broker endpoint. Using the registry descriptors and activity state, the broker forwards the CfP to potentially matching provider AASs. If dataspace communication is used, the registry relies on pre-negotiated dataspace contracts with the corresponding provider endpoints (Figure 4 middle "loop"). Each provider evaluates the CfP against its advertised capabilities, skills, parameters, availability and access conditions. Feasible providers send proposals directly to the requester AAS, again preferably through EDC-governed routes if available.

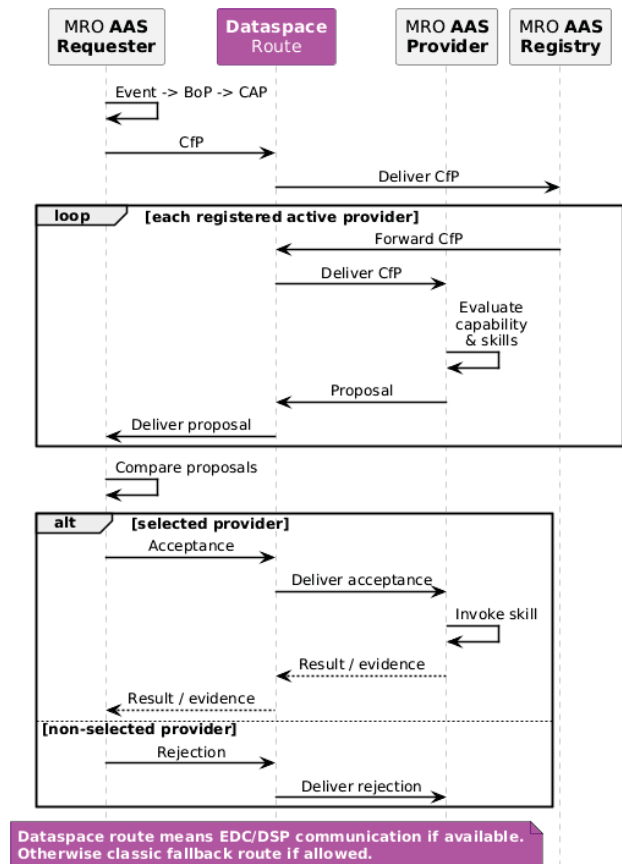


Figure 4: Dataspace-preferred CfP Choreography

The requester AAS evaluates the received proposals, selects the most suitable provider and sends an acceptance

document to the selected provider and rejection documents to the remaining providers (Figure 4 bottom). The accepted provider then executes the corresponding skill and returns the result, artefact or evidence reference. Thus, the AAS layer remains responsible for semantic service interaction, including CfPs, proposals, decisions, invocations and evidence references, while the dataspace layer governs selected message paths through contracts, policies and controlled endpoint access.

It is important to distinguish between AAS and EDC assets. An AAS asset is the represented physical or logical entity, such as an aircraft, component, registry, inspection station or service application. An EDC asset represents governed access to a selected endpoint of that entity, such as an AAS REST interface, an "/i40commu" path, a registry interface, a service API or an evidence repository. This separation enables the same AAS-based service choreography to operate through either direct local communication or sovereign cross-organizational dataspace routes.

5. AVIATION DATASPACE DEMONSTRATOR

This chapter verifies the requester–registry/broker–provider interaction, capability matching and EDC-governed invocation within the DLR-MO laboratory environment, while Figure 5 places this implemented path in the broader aviation MRO ecosystem. The demonstrator represents a distributed aviation MRO ecosystem rather than one predefined application chain. It combines requester assets, an AAS registry, physical, digital and hybrid service providers and participant-local dataspace components, while depicting potential evidence-consuming stakeholders. Figure 5 presents the overall ecosystem and its service choreography, while Table 2 and Table 3 document the implemented provider-side skills and requester-side Bill of Process (BoP)

variants. The Holybro X500 propeller inspection subsequently demonstrates the execution of one configured BoP through capability-based tendering, provider selection and dataspace-governed service invocation.

5.1. DLR Aviation MRO Ecosystem and Demonstrator Setup

The demonstrator reproduces the logical boundaries of a cross-organizational aviation MRO ecosystem through separate assets, applications, AAS instances, participant identities, endpoints, policies and EDC deployments. Although the physical implementations are located within the same laboratory environment, they operate as logically independent dataspace participants. A Holybro X500 unmanned aircraft system serves as an experimental surrogate for aircraft, subsystems, components and related aviation technologies. Its proactive AAS acts as the technical service requester on behalf of the operator, while the operator remains responsible for operational objectives, admissible service conditions and evidence requirements.

Figure 5 summarizes the ecosystem and illustrates a representative capability-based interaction. The upper part shows the stakeholder and lifecycle context: an aviation asset and its operator initiate a service request, several MRO providers offer alternative realizations, and an authority or another stakeholder is shown as a potential consumer of selected evidence. The sequence at the top structures the interaction into request, MRO execution, evidence provision and completion. The middle part represents the dataspace layer introduced in Chapter 3. Separate contracts govern the requester-to-registry, registry-to-provider, provider-to-requester and requester-to-selected-provider communication paths. Each contract defines the accessible endpoint, usage policy and associated access conditions.

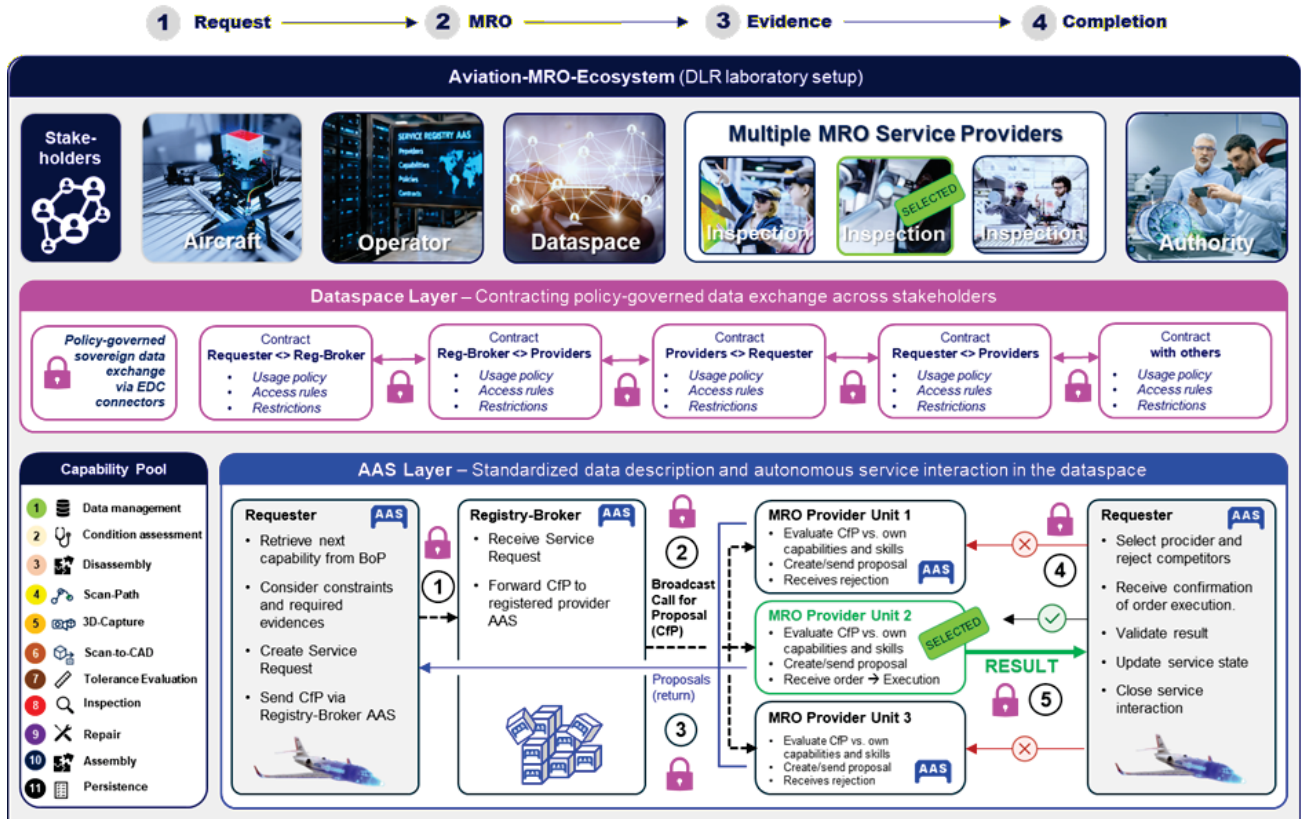


Figure 5: DLR aviation MRO ecosystem with dataspace-governed service interaction, illustrated by a capability-based inspection request

MRO Service	Capability	DLR App	Realized by skill	Input (examples)	Output (examples)
Inspection	Scan-Path	rayPlan	Ray-tracing-based visibility and scan-path planning	Reference CAD, scan area, robot/camera constraints	Scan poses, trajectory/path file
	3D-Capture	BotAssist	Robot-guided camera scan execution	Scan trajectory, robot configuration, camera settings	Image set, raw scan data, point cloud
	Scan-to-CAD	RGM	Point-cloud processing and surface reconstruction	Point cloud, initial/reference CAD, reconstruction parameters	Smoothed CAD, reconstructed geometry, point-cloud artefact
	Tolerance Evaluation	VTE	CAD-to-CAD comparison and deviation calculation	Reference CAD, scanned/reconstructed CAD, tolerance parameters	Deviation map, tolerance values, assessment result
Assessment	Condition assessment	BayCA	Bayesian probabilistic condition assessment	Sensor values, condition indicators, inspection results	Condition probability, confidence, anomaly indication, advisory
	Economic and Ecological assessment	LYFE	Dynamic cash-flow and inventory simulation	Aircraft, operator, manufacturer data, economic parameters, planning data	Absolute and delta economic values such as NPV, IRR
Persistence	Data management	FileHub	File-based artefact storage and provision	CAD files, point clouds, reports, images, metadata	File reference, evidence package, access link
	Image standardization	DICON	DICONDE metadata mapping and file conversion	Inspection images, metadata, asset/component identifiers	DICONDE file, standardized inspection metadata

Table 2: Demonstrator applications as example skill implementations for selected capabilities

Figure 5 additionally foresees a contract through which selected evidence can be made available to an authority or quality organization without involving that stakeholder in the preceding provider-selection process.

The lower part in Figure 5 represents the semantic and operational AAS layer introduced in Chapter 4. The capability pool on the left contains reusable functional building blocks that can be combined into different MRO services. In the centre, the requester AAS derives a capability need from its BoP and sends a CfP to the registry-integrated broker. The registry identifies registered and active providers, while the broker distributes the request to provider AASs with potentially compatible capability descriptions. Each provider evaluates the request against its available capabilities, skill implementations and service conditions and returns either a proposal or a rejection. The requester compares the received proposals, selects Provider Unit 2, rejects the remaining offers and invokes the selected skill. The requester representation on the right receives the result and evidence, validates the outcome, updates the service state and completes or continues the BoP.

Figure 5 therefore shows both the general DLR aviation MRO ecosystem and one exemplary inspection interaction within it. The architecture is not restricted to inspection; the same requester–registry–provider pattern is used for the additional services and capability combinations documented below.

5.2. Implemented Service Capabilities and BoP Variants

The demonstrator implements complementary provider-side and requester-side views. Table 2 presents the provider-side skill portfolio. Each physical, digital or hybrid resource is coupled with an AAS that describes its offered capability, associated skill, required inputs, generated outputs, parameters and invocable interfaces. The requester

can therefore evaluate the provider through a shared semantic description without requiring knowledge of its internal application, equipment or organizational realization.

For inspection, rayPlan implements scan-path planning, BotAssist robot-guided 3D data acquisition, RGM scan-to-CAD processing and VTE geometric tolerance evaluation. BayCA provides probabilistic condition assessment, while LYFE supports economic and ecological assessment. FileHub manages input artefacts, intermediate results and evidence references, and DICON converts inspection images and metadata into a standardized DICONDE representation. These applications constitute reusable skill implementations; they do not form one mandatory serial application chain.

The applications follow the MX-Port-based integration introduced in Section 3.3. Application- or equipment-specific interfaces are connected at the adapter layer, inputs and outputs are mapped to AAS semantic structures at the converter layer, and AAS operations, communication endpoints or service APIs provide the gate layer. For cross-organizational invocation, selected endpoints are additionally represented as EDC assets and exposed through associated contract offers and usage policies. The AAS provides the engineering semantics and service logic, while the EDC governs access to the corresponding endpoint.

Table 3 complements this provider-oriented implementation view with the requester and process perspective. Its rows represent configured MRO service variants, while its columns represent capabilities from the shared capability pool. A number indicates that a capability is part of the respective BoP and specifies its relative execution order. The matrix contains advisory, inspection, repair, assembly, persistence and overhaul variants and therefore documents that the demonstrator supports more than one service scenario.

Capability Pool (examples)

												
		Data management	Condition assessment	Economic assessment	Disassembly	Scan-Path	3D-Capture	Scan-to-CAD	Tolerance Evaluation	Repair	Assembly	Image standardization
MRO Services	Order specification	Bill of Process as sequenced capabilities										
Advisory	Condition, Eco KPI	3	1	2								
Inspection	basic	5				1	2	3	4			
Inspection	with disassembly	6			1	2	3	4	5			
Repair	with disassembly + inspection	8	1		2	3	4	5	6	7		
Assembly	with documentation	6					2	3	4		1	5
Inspection	with advisory	7	5	6		1	2	3	4			
Persistence	DICONDE-based	1, 3										2
Overhaul	full service	11	8	9	1	2	3	4	5	6	7	10

Table 3: Dynamic composition of service-specific BoP variants from a shared capability pool

The matrix also shows that capabilities are reusable across different service compositions. For example, Scan-Path planning, 3D-Capture, Scan-to-CAD processing and Tolerance Evaluation are used in the basic inspection BoP, but can also contribute to inspection with disassembly, repair or overhaul. Condition and economic assessment can be combined into an advisory service, while image standardization and data management support evidence-oriented persistence. Table 2 thus documents which provider-side skill realizations are available, whereas Table 3 shows how requester-side BoPs combine the corresponding capabilities into service-specific process variants.

The Inspection basic row is used for the propeller example in Section 5.3. It sequences scan-path planning, 3D-captures, scan-to-CAD processing and tolerance evaluation, followed by data management for the generated artefacts and evidence. Other rows demonstrate how the same capability pool can be extended by disassembly, advisory, repair, assembly or documentation activities.

5.3. Flexible Service Choreography

The exemplary interaction begins when the Holybro X500 condition-monitoring function detects elevated vibration in one propulsion unit. Such behaviour may indicate propeller imbalance, geometric damage or an anomaly in the associated drivetrain. The requester AAS associates the event with the affected propeller and evaluates the Inspection basic BoP from Table 3. For the demonstrated run, its subordinate capabilities are aggregated into the top-level request Inspection; at finer granularity, they may instead be tendered and assigned individually. The requester creates a CfP containing the propeller identifier, condition information, reference geometry, mandatory geometric properties, expected outputs and evidence requirements. The implementation-independent capability is refined through semantically identified properties as introduced in Section 4.3. The requester specifies admissible limits, whereas each provider advertises the values that can be achieved or verified by its provider-specific skill.

Three provider units advertise the same top-level inspection capability but offer different realizations. Unit 1 represents an inspection setup with limited geometric resolution and coverage. Unit 2 provides full-field optical 3D inspection with sufficient resolution and verifiable geometric results. Unit 3 provides high-resolution local analysis but cannot cover the complete defined inspection region. In this run, the provider units are alternatives for the aggregate inspection effect; with finer decomposition, different providers may execute consecutive BoP steps.

Table 4 represents the reduced digital tender document used for property-level feasibility evaluation. The requirements are derived from the demonstrator-specific quality inspection sheet and follow the modelling concepts of the Quality Control for Machining submodel [23]. They are example values and do not constitute certified manufacturer or airworthiness limits.

The comparison demonstrates that compatibility at capability-identifier level is insufficient. Although all three units advertise Inspection, only Unit 2 satisfies all mandatory request properties. Unit 1 cannot verify the required profile deviation and does not meet the required tip-length deviation, point spacing or surface coverage. Unit 3 achieves sufficient local point spacing but does not provide the full-field geometric measurements required by the tender. Unit 2 is therefore selected.

The CfP and proposals are exchanged through the requester–registry–provider choreography shown in Figure 5. Before a message crosses a participant boundary, the proactive AAS checks whether a valid contract agreement and usable endpoint data reference are available, as described in Section 4.5. Existing agreements can be reused; otherwise, the participant-local EDC performs catalogue retrieval, contract negotiation and transfer initiation. The registry-integrated broker forwards the CfP to the candidate providers, and the proposals are returned directly to the requester through the corresponding governed routes.

After evaluating the proposals, the requester sends an acceptance to Unit 2 and rejection messages to Units 1 and

3. The selected provider executes its inspection skill and returns the result, generated artefacts and evidence references. Depending on the concrete realization, these may include measurement values, images, point clouds, reconstructed geometry, geometric deviations, inspection reports and standardized metadata. The requester validates the returned outputs, updates the condition and service state of the UAS and evaluates the subsequent BoP transition. If no relevant defect is detected, the interaction is completed. Otherwise, the requester can instantiate a further BoP variant for additional analysis, repair, component replacement or re-inspection.

Service	Capability „Inspection“			
	Properties			
	profileDeviation [mm]	tipLengthDeviation [mm]	pointSpacing [mm]	surfaceCoverage [%]
Requirements				
Requester	≤ 0.50	≤ 1.00	≤ 0.20	≥ 95% of defined blade inspection region
Provision				
Unit 1 X	Not Verifiable X	Verifiable at ≤ 2.00 X	0.30 X	90 % X
Unit 2 ✓	Verifiable at ≤ 0.50 ✓	Verifiable at ≤ 0.50 ✓	0.10 ✓	98 % ✓
Unit 3 X	Not supported X	Not supported X	0.05 ✓	Only local region X

Table 4: Requested inspection properties and provider feasibility for the Holybro X500 propeller (reduced example)

Service granularity remains configurable. Supporting activities may be tendered as separate BoP steps or aggregated within a broader provider offer, provided that scope, preconditions, outputs, evidence obligations and responsibility boundaries remain explicit. The laboratory implementation verified the concept developed in the preceding chapters. Regarding RQ1, it showed that different aviation MRO services can be composed from reusable capabilities and realized through alternative provider-specific skills. Regarding RQ2, it demonstrated that registry/broker-mediated provider discovery, communication and endpoint access can be governed through dataspace contracts and policies without transferring semantic service logic to the connector. Regarding RQ3, it verified that proactive AAS can represent requester and provider assets, derive service needs from a BoP, perform property-level capability matching, select and invoke a provider-specific skill and process the resulting evidence. The demonstrator therefore provides an integrated technical verification of the implemented requester–

registry/broker–provider path within the proposed AAS- and dataspace-based aviation MRO service ecosystem.

6. CONCLUSION

This paper developed and technically verified an aviation dataspace architecture for sovereign, capability-based MRO and manufacturing collaboration. Its central contribution is the separation and integration of three concerns: BoPs and shared AAS capability descriptions define service demand and provider-independent semantics; proactive AAS perform event-driven, property-level service negotiation and invoke provider-specific skills; and EDC-based dataspace mechanisms govern cross-organizational access through identities, contracts, policies and controlled transfer. In the DLR demonstrator, alternative inspection providers were evaluated, the only unit satisfying all mandatory propeller-inspection properties was selected, and the resulting evidence was returned to the requester.

The results show that semantic interoperability, autonomous technical service interaction and data sovereignty can be combined without centralizing raw data or embedding domain logic in the connector. Further work will address scalability, dynamic trust and contract handling, certification-relevant evidence chains and deployment across independent industrial organizations.

7. ACKNOWLEDGEMENT

The authors gratefully acknowledge the support from the German Federal Ministry for Economic Affairs and Energy (BMWE) through the Aerospace-X project (Grant No. 13MX004B) as part of the Manufacturing-X initiatives. In addition, we would like to thank our partners in the project for the fruitful collaboration.

8. ABBREVIATIONS

AAS	Asset Administration Shell
ADS	Asset Data Server
AID	Asset Interface Description
API	Application Programming Interface
BoP	Bill of process
CAD	Computer Aided Design
CAMO	Continuing Airworthiness Mgmt. Organisation
CfP	Call for Proposal
CPS	Cyber-Physical System
CSS	Capability-Skill-Service
CX	Catena-X
DCP	Decentralized Claims Protocol
DLR	Deutsches Zentrum für Luft- und Raumfahrt e.V.
DPP	Digital Product Passport
DSP	Dataspace Protocol
EDC	Eclipse Dataspace Connector
I4.0C	Industry 4.0 Component
I4.0CS	Industry 4.0 Component Stack
IDSA	International Data Spaces Association
IDTA	Industrial Digital Twin Association e.V.
IIoT	Industrial Internet of Things
IoT	Internet of Things
IT	Information Technology
MaSiMO	Maintenance Simulation Model @ DLR MO
MQTT	Message Queuing Telemetry Transport
MRO	Maintenance, Repair and Overhaul
NDT	Non-Destructive Testing
OEM	Original Equipment Manufacturer
OPC UA	Open Platform Comm. Unified Architecture
PHM	Prognostics and Health Management

REST Representational State Transfer
 SM Submodel
 UAS Unmanned Aircraft System

9. REFERENCES

- [1] Aerospace-X Consortium. Building a digital ecosystem for an efficient and sustainable Aerospace Supply Chain, Project Framework and Vision, Manufacturing-X Initiative, **2024**, <https://www.aerospace-x.net> (accessed on 12.07.2026)
- [2] Catena-X Automotive Network e.V. Catena-X. Available online: <https://catena-x.net/en/> (accessed on 26.06.2026)
- [3] DECADE-X Association. DECADE-X: Industry Gets Organized for the Supply Chain Digitalization, Official Foundation and Governance Framework, Paris, France, **2025**, Available: <https://www.decade-x.org> (accessed on 12.07.2026)
- [4] Diedrich, C.; Belyaev, A.; Hermann, J. et al. Information Model for Capabilities, Skills & Services. Plattform Industrie 4.0: Berlin, Germany, **2022**. Available online: <https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/CapabilitiesSkillsServices.pdf> (accessed on 15.06.2026)
- [5] DIN EN IEC 63278-1 VDE 0810-781:2022-07 Asset Administration Shell for industrial applications, **2022**, VDE Standards, VDE Verlag, Berlin
- [6] Eclipse Dataspace Components. IdentityHub: Decentralized Claims Protocol Wallet and Verifiable-Credential Management. Eclipse Foundation, software architecture and implementation documentation (accessed on 12.07.2026)
- [7] Eclipse Tractus-X. Tractus-X EDC: Control-Plane and Data-Plane Distributions. Eclipse Foundation, software architecture and implementation documentation (accessed on 12.07.2026)
- [8] Factory-X Consortium. The MX-Port Concept: A New Foundation for Cross-Industry Data Sharing, White Paper, Version 1.0, Factory-X / Manufacturing-X Initiative, Frankfurt am Main, Germany, **2025**, <https://factory-x.org/factory-x-introduces-the-mx-port-concept> (accessed on 12.07.2026)
- [9] Factory-X. The Digital Ecosystem. Available online: <https://factory-x.org> (accessed on 26.06.2026)
- [10] Gabellini, M.; Civolani, L.; Ronchi, M.; Naldi, L.D.; Regattieri, A. Data Spaces in Manufacturing and Supply Chains: A Review and Insights from European Initiatives, Article, **2025**, Appl. Sci., 15, 5802. DOI: 10.3390/app15115802
- [11] Ganser, P.; Weiss, M. et al. Data Spaces As a Foundation for End-to-End Quality Management in Aerospace Manufacturing, 2026, ASME Turbomachinery Technical Conference, Milan, Italy, accepted for publication
- [12] Grunau, S.; Redeker, M.; Göllner, D.; Wisniewski, L. The Implementation of Proactive Asset Administration Shells: Evaluation of Possibilities and Realization in an Order Driven Production. **2022**, In: Jasperneite, J., Lohweg, V. (eds) Kommunikation und Bildverarbeitung in der Automation. Technologien für die intelligente Automation, vol 14. Springer Vieweg, Berlin, Heidelberg
 DOI: 10.1007/978-3-662-64283-2_10
- [13] Gupta, P. K.; Heller, J.; Lüdemann-Ravit, B. Asset Administration Shell: Deutschlands Standard für Digitale Zwillinge, Article, **2025**, ZWF Zeitschrift für Wirtschaftlichen Fabrikbetrieb (120)
 DOI: 10.1515/zwf-2025-1121
- [14] HashiCorp. Vault Documentation: Secrets, Cryptographic Material and Identity-Based Access. HashiCorp (accessed on 12.07.2026)
- [15] IDTA 01001 Specification of the Asset Administration Shell Part 1: Metamodel, Specification, Version 3.1.2, **2025**, Industrial Digital Twin Association, DOI: 10.62628/IDTA.01001-3-1-2
- [16] IDTA 01004 Specification of the Asset Administration Shell Part 4: Security, Version 3.0.2, **2026**, publisher: Industrial Digital Twin Association, Frankfurt am Main, Germany, DOI: 10.62628/IDTA.01004-3-0-2
- [17] IDTA 02010-1-0 Service Request Notification, Submodel Template of the AAS, **2023**, publisher: Industrial Digital Twin Association, Frankfurt am Main, Germany, <https://industrialdigitaltwin.org> (accessed on 04.07.2026)
- [18] IDTA 02015-1-0, Control Component Type, Submodel Template of the Asset Administration Shell, **2023**, publisher: Industrial Digital Twin Association, Frankfurt am Main, Germany, available online: <https://industrialdigitaltwin.org> (accessed on 04.07.2026)
- [19] IDTA 02016-1-0, Control Component Instance, Submodel Template of the Asset Administration Shell, **2023**, publisher: Industrial Digital Twin Association, Frankfurt am Main, Germany, available online: <https://industrialdigitaltwin.org> (accessed on 04.07.2026)
- [20] IDTA 02017-1-0 Asset Interface Description, **2024**, publisher: Industrial Digital Twin Association, Frankfurt am Main, Germany, <https://industrialdigitaltwin.org> (accessed on 04.07.2026)
- [21] IDTA 02020 Capability Description, Submodel Template of the AAS, **2026**, publisher: Industrial Digital Twin Association, Frankfurt am Main, Germany, <https://industrialdigitaltwin.org> (accessed on 04.07.2026)
- [22] IDTA 02027-1-0 Asset Interfaces Mapping Configuration, **2024**, publisher: Industrial Digital Twin Association, Frankfurt am Main, Germany, <https://industrialdigitaltwin.org> (accessed on 04.07.2026)
- [23] IDTA 02049 Quality Control for Machining, Submodel Template of the AAS, **2026**, publisher: Industrial Digital Twin Association, Frankfurt am Main, Germany, <https://industrialdigitaltwin.org> (accessed on 12.07.2026)
- [24] IDTA 02099-1 Digital Product Passport – Part 1, 2026, publisher: Industrial Digital Twin Association, Frankfurt am Main, Germany, <https://industrialdigitaltwin.org> (accessed on 13.07.2026)
- [25] Jacoby, M.; Jovicic, B.; Stojanovic, L.; Stojanovic, N. An Approach for Realizing Hybrid Digital Twins Using Asset Administration Shells and Apache StreamPipes, Article, Information, **2021**, 12, 217. DOI: 10.3390/info12060217
- [26] Jacoby, M.; Volz, F.; Weißenbacher, C.; Müller, J. FA³ST Service – An Open Source Implementation of the Reactive Asset Administration Shell, **2022** DOI: 10.1109/ETFA52439.2022.9921584.
- [27] Kagermann, H.; Wahlster, W.; Lukas, W.-D. Recommendations for Implementing the Strategic Initiative INDUSTRIE 4.0: Final Report of the Industrie 4.0 Working Group, Report, **2013**, Forschungsunion/acatech – National Academy of Science and Engineering
- [28] Lee, J.; Bagheri, B.; Kao, H.-A. A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems, Journal Article, **2015**, Manufacturing Letters, Volume 3, Pages 18-23, DOI: 10.1016/j.mfglet.2014.12.001

- [29] Nagel L.; Lycklama D. Design Principles for Data Spaces. Position Paper, **2021**, Version 1.0. Berlin, DOI: 10.5281/zenodo.5105744
- [30] Prentice, B. et al. Global Fleet and MRO Market Forecast 2026–2036, **2026**, Oliver Wyman, <https://www.oliverwyman.com/content/dam/oliverwyman/v2/publications/2026/february/global-fleet-and-mro-market-forecast-2026-2036.pdf> (accessed on 15.07.2026)
- [31] Schöppenthau, F. et al. Building a Digital Manufacturing as a Service Ecosystem for Catena-X, **2023**, Sensors, 23(17), 7396 DOI:10.3390/s23177396
- [32] The Eclipse Foundation. Eclipse BaSyx: Industry 4.0 Operating System. <https://eclipse.dev/basyx> (accessed on 15.06.2026)
- [33] Urban, C.; Belyaev, A.; Diedrich, C. Verwaltungsschale-basierter Ansatz für die Umsetzung von auftragsgesteuerter Produktion, **2022**, Conference paper, 23. VDI-Kongress AUTOMATION – Leitkongress der Mess- und Automatisierungstechnik, Baden-Baden, Germany
- [34] VDI/VDE 2193 Part 1:2020-04 Language for I4.0 Components - Structure of messages, VDI/VDE guideline **2020**, VDI-VDE Manual, Düsseldorf
- [35] VDI/VDE 2193 Part 2:2020-01 Language for I4.0 components - Interaction protocol for bidding procedures, VDI/VDE guideline **2020**, VDI-VDE Manual, Düsseldorf
- [36] Vieira da Silva, L. M.; Köcher, A.; Gill, M.; Weiss, M.; Fay, A. Toward a Mapping of Capability and Skill Models using Asset Administration Shells and Ontologies, Paper, **2023**, IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA) DOI: 10.1109/ETFA54631.2023.10275459
- [37] Weiss, M.; Grishchenko, E.; Raddatz et al. MaSiMO – Autonomous Quality Inspection in Aviation MRO and Manufacturing Using Asset Administration Shell, Digital Product Passport and Eclipse Dataspace Connector, **2025**, Paper, Deutsche Gesellschaft für Luft- und Raumfahrt - Lilienthal-Oberth e.V. DOI: 10.25967/650135.
- [38] Weiss, M.; Raddatz, F.; Wende, G. MaSiMO – Digital Product Passport and autonomous event management of Industry 4.0 Components with proactive AAS in data-driven aviation maintenance and production, **2024**, Paper, Deutsche Gesellschaft für Luft- und Raumfahrt - Lilienthal-Oberth e.V. DOI: 10.25967/630084
- [39] Weiss, M.; Raddatz, F.; Wende, G. Predictive Maintenance for Life Cycle Engineering Using I4.0 Technologies in MRO Data Systems, Paper, **2026**, Procedia CIRP (140), ISSN: 2212-8271 DOI: 10.1016/j.procir.2026.05.114
- [40] Weiss, M.; Wicke, K.; et al. MaSiMO - Development and Research of Industry 4.0 Components with a Focus on Experimental Applications of Proactive Asset Administration Shells in Data-Driven Maintenance Environments, **2023**, Paper, Deutsche Gesellschaft für Luft- und Raumfahrt - Lilienthal-Oberth e.V. DOI: 10.25967/610125
- [41] Weiss, M.; Wicke, K.; Wende, G. MaSiMO - A Hybrid Experimental Platform for the Simulation and Evaluation of Data-Driven Maintenance Enterprises, **2022**, Paper, Deutsche Gesellschaft für Luft- und Raumfahrt - Lilienthal-Oberth e.V. DOI: 10.25967/570154
- [42] Winkler, D.; Gill, M. S.; Fay, A. The Asset Administration Shell as a solution concept for the realisation of interoperable Digital Twins of Aircraft Components in Maintenance, Repair and Overhaul, **2022**, Paper, Deutsche Gesellschaft für Luft- und Raumfahrt - Lilienthal-Oberth e.V. DOI: 10.25967/570274
- [43] Zhang, J., Ellwein, C., Heithoff, M. et al. Digital twin and the asset administration shell, Article, **2025**, Softw Syst Model 24, 771–793, DOI: 10.1007/s10270-024-01255-0