Project title	**ACHILLES Human-Centred Machine learning lighter, clearer, safer**		
Project acronym	ACHILLES		
GA number	101189689		
Project start date	01/11/2024	Duration	48 months

D1.1 - DATA STANDARDISATION AND INTEROPERABILITY GUIDELINES

Due date	30/04/2026	Delivery date	30/04/2026
Work package	WP1		
Responsible Author(s)	Sînică Alboaie (AXL)		
Contributor(s)	Nicoleta Mihalache (AXL), Alexandrina Rata (AXL), Julie Mannekens (KLU), André Carreiro (FhAICOS), Marco Cuomo (CuomoIT), Darlene MacDonald (CuomoIT)		
Reviewer(s)	Rui Castro (FhAICOS)		
Version	V1.0		
Dissemination level	Public		

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

VERSION AND AMENDMENTS HISTORY

Version	Date (DD/MM/YYYY)	Created /Amended by	Changes
V0.1	01/03/2026	Nicoleta Mihalache (AXL)	Integrated content prepared during M1-M16
V0.2	30/03/2026	Julie Mannekens (KLU) Sînică Alboaie (AXL)	Legal chapter
V0.3	30/03/2026	Sînică Alboaie (AXL) Alexandrina Rata (AXL)	Internal WP1 review and fixes
V0.4	01/04/2026	André Carreiro (FhAICOS)	Introduces Cardsmith
V0.5	06/04/2026	Rui Castro (FhAICOS)	Review
V0.6	16/04/2026	Marco Cuomo (CuomoIT)	Review & Rephrases
V0.7	24/04/2026	All	Adjustments after review
V1.0	29/04/2026	André Carreiro (FhAICOS)	Final review and clean-up

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

TABLE OF CONTENTS

Funded by the European Union
This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
AIDOC-AP	AI Documentation Application Profile
AIRO	AI Risk Ontology
API	Application Programming Interface
AsyncAPI	Async API Specification
CLI	Command Line Interface
DGA	Data Governance Act
DID	Decentralised Identifier
DPIA	Data Protection Impact Assessment
DPV	Data Privacy Vocabulary
DSU	Data Sharing Unit
D x.y	Deliverable x.y

Funded by the European Union
This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

ECDS	European Common Data Spaces
EHDS	European Health Data Space
EU	European Union
FRIA	Fundamental Rights Impact Assessment
GAIA-X	European federated data infrastructure initiative
IDE	Integrated Development Environment
JSON	JavaScript Object Notation
JSON-LD	JavaScript Object Notation for Linked Data
LLM	Large Language Model
MCP	Model Context Protocol
MCRO	Model Card Report Ontology
ML	Machine Learning
OpenAPI	Open API Specification
PROV	W3C Provenance Data Model
RDF	Resource Description Framework
RFC	Request for Comments
SLSA	Supply-chain Levels for Software Artefacts
SPDX	Software Package Data Exchange
WPx	Work Package x

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

1 EXECUTIVE SUMMARY

This deliverable, D1.1 – Data Standardisation and Interoperability Guidelines v1, defines the project-level interoperability foundation for ACHILLES. Its purpose is to establish the common structures needed for consistent integration, reuse, validation, and governance of data assets, AI artefacts, services, plugins, skills, and agent-usable capabilities across the project. D1.1 adopts a selective and pragmatic approach to standardisation. Rather than prescribing a rigid technical architecture, it defines a minimum interoperability layer that can support convergence across work packages while preserving flexibility for research, experimentation, and use-case-specific development. The deliverable focuses on standardising what is needed for cross-project coordination: capability descriptors, plugin manifests, metadata structures, provenance and lineage information, access and usage distinctions, runtime assumptions, validation evidence, and structured AI documentation.

A central contribution of D1.1 is the definition of how reusable capabilities should be exposed to the ACHILLES environment. APIs, services, tools, skills, and plugins intended for cross-work-package reuse or ACHILLES IDE integration should provide machine-readable and human-inspectable descriptors. These descriptors should explicitly specify the purpose, invocation mode, inputs, outputs, dependencies, permissions, side effects, provenance hooks, and validation requirements for each capability. This supports controlled discovery and invocation within the ACHILLES IDE and the broader multi-agent environment. The deliverable also defines a provenance-aware approach to interoperability. Shared datasets, derived artefacts, models, validation outputs, and governance documentation should carry sufficient metadata, lineage, integrity references, access conditions, and validation status to remain traceable across workflows. This supports data auditing, explainability, monitoring, compliance documentation, and trustworthy reuse. Structured AI documentation, including model cards, data cards, system cards, risk assessments, and compliance checklists, is treated as part of this interoperability layer rather than as a separate reporting activity.

D1.1 further prepares ACHILLES for future alignment with European Common Data Spaces and related data-governance frameworks. It does not implement a complete connector stack, but it identifies the architectural preconditions needed for future compatibility: persistent asset identification, metadata and provenance structures, access descriptors, integrity references, and governance-aware documentation.

The deliverable deliberately avoids premature standardisation in areas where technical choices depend on later implementation or validation work. It does not define a universal project ontology, a final Agent-to-Agent protocol, a federated-learning payload format, a hardware-specific model representation, or a complete European Data Space connector implementation. These remain under the responsibility of the relevant work packages and use-case validation activities.

The guidelines introduced in D1.1 apply when a component, dataset, model, service, plugin, agent, connector, or documentation artefact enters the shared ACHILLES interoperability layer, for

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

example through cross-work-package reuse, ACHILLES IDE integration, regulated data-sharing workflows, or common validation. Internal prototypes may continue to evolve locally until they are prepared for shared use. Overall, the deliverable provides a structured yet adaptable foundation for interoperability within the ACHILLES project. It establishes common reference points for collaboration, reduces integration risks, and creates the conditions for progressive evolution towards trustworthy, reusable, and policy-compliant AI-enabled data processing capabilities.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

2 INTRODUCTION

The purpose of D1.1 is to define the project-level interoperability layer for ACHILLES. Within the overall project architecture, this deliverable establishes the common structures required for cross-work-package integration, the operation of the ACHILLES IDE, and the controlled exposure of services, data assets, plugins, and agent-usable capabilities. The scope of the deliverable is deliberately focused. D1.1 standardises only those aspects of interoperability that are cross-cutting, recurrent, and enabling at the project level. These include capability contracts, provenance and lineage structures, minimum metadata requirements, access-relevant descriptors, structured AI documentation, and architectural assumptions needed for future alignment with European data-space environments. It does not attempt to define all domain semantics, runtime protocols, agent coordination mechanisms, or framework-specific payloads.

ACHILLES targets AI systems that are lighter, clearer, and safer. In this context, interoperability cannot be reduced to transport compatibility or the exchange of syntactically valid payloads. It also concerns whether a capability can be discovered and invoked under explicit assumptions; whether a dataset or derived artefact carries sufficient lineage information to remain trustworthy after transformation; whether access conditions remain visible across processing stages; and whether heterogeneous modules can be integrated without losing provenance, policy, or governance-relevant information [23]. These are project-level concerns because they affect several work packages simultaneously and condition the practical coherence of the integrated ACHILLES environment.

Task T1.1 and this deliverable D1.1, therefore, define the minimum common substrate needed for interoperability across the project. It does not define a universal project ontology, a final Agent-to-Agent protocol, a federated-learning payload format, a hardware-specific model representation, or a complete European Data Space connector implementation. Those areas remain under the responsibility of the work packages and validation settings where the relevant technical evidence, framework choices, and domain constraints are available. The practical applicability rule is simple: internal prototypes may remain flexible while they are confined to a local task or work package. Once a component, dataset, service, plugin, agent, or connector is intended for cross-work-package reuse, ACHILLES IDE integration, regulated data-sharing workflows, or common validation, it becomes part of the shared interoperability layer and should conform to the relevant D1.1 guidelines. The implementation of these guidelines remains distributed across the relevant work packages. WP1 defines the common interoperability expectations. WP2 implements privacy-preserving and locality-preserving computation mechanisms. WP3 implements advanced multi-agent orchestration and execution logic. WP6 will materialise many of these standards within the ACHILLES IDE through plugin exposure, capability discovery, and integration interfaces. WP7 will validate the adequacy of these structures in concrete use cases. D1.1 therefore defines the minimum common layer upon which these activities can rely.

Funded by the European Union
This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

2.1 Prioritisation of standardisation areas

The selective scope of D1.1 is reflected in Table 1. The table distinguishes between standardisation areas that require early project-level alignment and areas that should remain deferred until more technical or use-case-specific evidence is available.

Table 1. Comprehensive Assessment and Prioritisation

Standardisation Aspect & Priority Level	**Scope, Rationale & Project Alignment**
API & Agentic Skill Contracts (Priority: Very High)	Defines how modules expose functionalities to external callers, including AI agents. This includes capability descriptors and relevant protocol mappings, such as the Model Context Protocol (MCP). Rationale (Critical for WP6 & WP3): It enables local tools and endpoints to become reusable capabilities that can be discovered, invoked, validated, and orchestrated by the ACHILLES IDE and multi-agent environment.
Data Provenance, Lineage & Structured AI Documentation (Priority: High)	Defines the minimum tracking mechanisms needed to record the origin, transformations, custody, integrity references, and governance-relevant context of data assets, derived artefacts, models, and reusable capabilities. This also includes structured AI documentation artefacts such as model cards, data cards, compliance checklists, and related documentation templates. Rationale (Core to Trustworthy AI): Directly supports WP1 (Data Auditing) and WP5 (Explainable AI / Model Cards) via the Cardsmith framework. It is central for the "Clearer" and "Safer" pillars of ACHILLES.
European Data Spaces (ECDS) Connectors (Priority: High)	Defines the architectural patterns and metadata wrappers required to ingest data from, and share data with, European Common Data Spaces. Rationale (Strategic Alignment): Mandatory for the project's adherence to European guidelines and crucial for the success of highly regulated use cases that depend on sovereign data sharing.
Fundamental I/O Data Formats (Priority: Low / Deferred)	Establishes universally accepted formats for standard data types across the consortium (e.g., Parquet for tabular data, standard JSON structures for text, standard encodings for imagery). Rationale (Baseline Interoperability): Prevents integration bottlenecks between WP1 (Data-centric) and WP3 (Model-centric) operations.

Funded by the European Union
This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

	D1.1 therefore encourages clear format declaration and metadata description rather than imposing a universal set of formats at this stage.
Federated Learning Payload Structures (Priority: Low / Deferred)	Concerns the format of model weights, gradients, and aggregation protocols exchanged during decentralised training. Rationale (WP2 Dependency): Highly relevant for privacy-preserving ML, but standardisation should be led by the specific requirements of the Federated Learning frameworks chosen in WP2, rather than pre-emptively locked in WP1.
Bias and Fairness Annotations (Priority: Low / Deferred)	Concerns vocabulary for tagging datasets, models, or evaluation results with information about demographic, representational, or statistical bias. Rationale (Important but Evolving): Essential for T1.2, but the taxonomy of bias is highly context dependent. A flexible schema is required rather than a rigid, hardcoded standard at this stage.
Domain-Specific Ontologies (Priority: Low / Deferred)	Concerns strict semantic modelling for specific sectors or use cases (e.g., standards for pharmaceutical trials, or specific screenplay formatting schemas). Rationale (Use-Case Specific): Highly relevant to HERA and other use cases, but defining these globally in D1.1 creates unnecessary rigidity. These should be defined locally within the WP7 validation use cases.
Hardware-Specific Tensor/Quantisation Formats (Priority: Low / Deferred)	Concerns low-level model and tensor representations optimised for specific deployment targets (e.g., INT8, FP16 structures for edge inference). Rationale (Out of WP1 Scope): Belongs strictly to the deployment-centric phase (WP3). Imposing hardware-level data standards in WP1 would prematurely constrain model training and optimisation research.

The table shows that D1.1 focuses on areas where early convergence is necessary for project coherence. API and agentic skill contracts are in scope because they are required for the ACHILLES IDE, for WP3 multi-agent integration, and for the transformation of local tools into reusable capabilities. Provenance, lineage, and structured AI documentation are in scope because they support data auditing, explainability, validation, compliance reporting, and trustworthy reuse across workflows. ECDS-related metadata [43] and access descriptors are in scope because future interoperability with sovereign European data-sharing environments depends on these foundations.

Funded by the European Union
This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

By contrast, federated-learning payload structures, domain-specific ontologies, detailed bias taxonomies, and hardware-level optimisation formats are not standardised in D1.1. These are relevant technical topics, but their correct formulation depends on later implementation choices, local domain requirements, or framework-level constraints.

2.2 Interoperability in the context of Agentic AI

Interoperability has long been a foundational goal of digital systems, but its practical meaning is changing. At the syntactic level, interoperability depends on formats, schemas, endpoint descriptions, payload structures, serialisation conventions, and transport mechanisms. These remain necessary: without basic syntactic agreement, information exchange becomes unreliable. However, syntax is not sufficient. Two systems may parse the same structure and still fail to cooperate meaningfully because they attach different assumptions, classifications, risks, or intended uses to the same data.

This distinction is especially important for ACHILLES. The project is not building a conventional software stack in which systems only exchange static data. It is moving toward an environment in which AI components, agents, services, developers, and governance artefacts interact around dynamic, contextual, and increasingly executable resources. In this setting, interoperability must also support actionable intelligibility: an AI agent, plugin, service, or human operator should be able to understand what a resource is, what it is for, how it may be used, the assumptions that govern it, and the boundaries within which it can be trusted. This does not mean that the project should standardise everything. In a research project operating at the intersection of AI engineering, experimentation, and evolving use cases, premature standardisation can be harmful [1], risking freezing areas that should remain open to discovery. The appropriate strategy is therefore selective: standardise the stable, cross-cutting core, while allowing local translation, adaptation, and experimentation to continue where requirements are still evolving.

Large Language Models (LLMs) and AI agents change the economics of integration, but they do not remove the need for explicit standards. LLMs can help translate between schemas, infer approximate correspondences, generate adapters, normalise documentation, and support semantic mapping across heterogeneous resources [5, 26]. This is valuable, especially in a project where use cases and technical components are diverse. However, LLM-based mediation is probabilistic and context-sensitive [24]. It may collapse ambiguity too early, produce plausible but insufficiently justified mappings, or hide differences that should remain visible. For this reason, LLM-based mediation should be treated as a useful capability bounded by explicit interfaces, provenance, review points, and, where necessary, symbolic or contractual validation.

This is also why, in D1.1, structured documentation is treated as part of interoperability rather than an administrative afterthought. Governance artefacts such as model cards, data cards, risk assessments, compliance checklists, and validation reports must be traceable, comparable, and reusable across work packages. Cardsmith and related documentation mechanisms can support this goal by helping transform documentation into structured, repeatable, and interoperable project

Funded by the European Union
This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

artefacts. In this sense, documentation becomes part of the interoperability layer: it carries information about assumptions, constraints, evidence, risk, and intended use.

2.3 Position on protocols and agent-to-agent interoperability

D1.1 gives particular attention to capability exposure and agentic integration. The Model Context Protocol (MCP) [22] is relevant because it defines a practical approach for AI systems to access tools, resources, prompts, and contextual services across boundaries. It is therefore directly aligned with the ACHILLES ambition to transform conventional endpoints into machine-usable capabilities that can be integrated by agents and copilots [28]. However, MCP should not be treated as a universal solution to interoperability. It is best understood as a practical standard for one strategically important layer: the interface between agentic runtimes and external capabilities. Within ACHILLES, this makes MCP and related interface-description approaches high-priority infrastructural concerns, but it does not imply the standardisation of all higher-level semantics, domain ontologies, or coordination patterns.

At this stage, ACHILLES does not propose a definitive project-wide Agent-to-Agent communication protocol. The project adopts a research-guided position informed by previous work on executable choreographies [3, 4] and by the ongoing evolution of industry and academic practice. Free-form natural language will remain useful for flexible coordination between agents, but it is unlikely to be sufficient on its own where traceability, bounded autonomy, reproducibility, and auditability are required. For that reason, Agent-to-Agent interoperability [18] is treated as a strategically important layer that may require lightweight explicit communication structures, especially for delegation, reporting, validation, escalation, and artefact exchange. The working position of D1.1 is therefore intermediate. Interoperability in the era of agentic AI should not be reduced to either traditional format standardisation or the assumption that LLMs will solve semantic alignment dynamically. Modern AI adds an adaptive layer of semantic mediation, but that layer becomes trustworthy only when bounded by explicit contracts, provenance, validation evidence, and governance-aware interfaces.

2.4 How to read this deliverable

The document should be read in two layers. Chapters 3 to 6 provide the rationale and architectural orientation for the ACHILLES interoperability approach. Annex A translates this orientation into operational requirements, compliance expectations, indications of implementation ownership, and minimum validation evidence. The main body explains why the selected standardisation areas matter, while the annex clarifies how partners should apply them when components enter the shared project environment. D1.1 therefore defines the common interoperability foundation for ACHILLES: instead of a rigid, universal architecture, it establishes a practical set of project-level guidelines that support integration, traceability, reuse, validation, and alignment with European data-sharing environments.

Funded by the European Union
This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

3 UNDERLYING LEGAL FRAMEWORK

This chapter identifies the legal and policy context that is directly relevant to the interoperability guidelines defined in D1.1. Its purpose is not to provide a general legal analysis of data governance, nor to replace the more comprehensive assessment provided in D4.1 – Legal and Ethical Mapping. Rather, it extracts the interoperability-relevant implications of selected EU frameworks and explains how they inform the technical and architectural choices made in this deliverable.

D1.1 defines project-level structures for cross-work-package integration, including capability descriptions for APIs, skills, plugins, and reusable capability packages; provenance-carrying metadata; minimum access semantics; structured documentation artefacts; and architectural assumptions for future alignment with European data-space environments. Beyond technical conveniences, these structures also support broader European policy objectives around data availability, trusted sharing, sovereignty, traceability, and accountable reuse.

Three legal and policy references are particularly relevant for this deliverable. First, the European Data Strategy and the Data Union Strategy provide the strategic policy context for interoperable data sharing in Europe [41]. Second, the EU Data Act introduces specific interoperability requirements for data, data-sharing mechanisms, and related services [42]. Third, the development of Common European Data Spaces [43] provides the broader governance and infrastructure model with which ACHILLES should remain compatible as the project evolves.

3.1 European Data Strategy and Data Union Strategy

The European Data Strategy frames interoperability as a condition for creating a functioning single market for data. Its objective is to enable data to flow across sectors and Member States while preserving European values, data protection, and the rights of individuals and organisations that generate or control data [40]. Within this strategy, Common European Data Spaces are presented as a central mechanism for making more data available for reuse under trusted and controlled conditions.

The Data Union Strategy builds on this direction by emphasising the availability of high-quality data for AI and innovation, the simplification of data-related rules, and the strengthening of Europe's position in international data flows [41]. For D1.1, the most relevant aspect is the continued emphasis on interoperable data-sharing environments. This supports the need for ACHILLES to define common metadata, provenance, access, and capability-description structures early in the project, even before full operational integration with external data spaces is implemented. The implication for ACHILLES is that interoperability should not be treated only as a technical matter of formats and interfaces. It should also support trusted reuse, data sovereignty, transparency, and controlled access. This is consistent with the project's objective of developing AI systems that are lighter, clearer, and safer, and with the role of D1.1 as the project-level guideline for consistent and interoperable data usage.

Funded by the European Union
This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

3.2 Data Act interoperability requirements

The Data Act, which took effect on 12 September 2025, aims to ensure the fair distribution of data value by establishing rules to facilitate data access and data use within the EU [42]. Its primary objective is to enhance data availability, promote fair access, and protect user rights while ensuring the protection of personal data. It is cross-sectoral legislation and applies across all sectors. For D1.1, the most relevant part of the EU Data Act is Chapter VIII, which addresses data interoperability, data-sharing mechanisms, and related services. Article 33 is particularly important because it links interoperability to several requirements directly relevant to the ACHILLES interoperability layer.

In practical terms, Article 33 points to the need for data and data-sharing mechanisms to be described in ways that allow recipients to find, access, understand, and use data. This includes information on dataset content, use restrictions, licences, collection methodology, data quality, uncertainty, data structures, formats, vocabularies, classification schemes, taxonomies, code lists, technical access mechanisms, API descriptions, terms of use, and quality of service. These elements closely correspond to the metadata, access, provenance, and capability-description concerns addressed in this deliverable. D1.1 therefore uses Article 33 as a legal and policy reference point for defining the minimum interoperability expectations within ACHILLES. The project does not need to replicate the Data Act as a legal checklist in this deliverable. Instead, it translates the relevant interoperability principles into project-level guidelines: shared descriptors, machine-readable metadata, provenance information, access-relevant declarations, and technical descriptions that support controlled reuse across work packages and, later, potential data-space integration.

Article 35 of the Data Act concerns interoperability of data-processing services. At this stage, this provision does not appear to impose direct obligations on ACHILLES, as the project is not intended to provide data-processing services within the meaning of the Data Act. Its relevance should nevertheless be reassessed if later project outcomes evolve into externally provided cloud, edge, or managed data-processing services.

Similarly, Article 36 concerns smart contracts used to automate the execution of data-sharing agreements. This provision does not appear directly applicable at this stage, as ACHILLES does not currently foresee the deployment of smart contracts for that purpose. If later connector or data-space integration work introduces automated data-sharing agreement mechanisms, the requirements of Article 36 should be revisited.

3.3 Common European Data Spaces

Common European Data Spaces are a key part of the European approach to trusted data sharing. They aim to create sector-specific environments in which data can be pooled, accessed, shared, processed, and reused under fair, transparent, proportionate, and non-discriminatory conditions, while respecting applicable EU law [43]. By harmonising rules on data interoperability, it is hoped that these data spaces will operate more efficiently, enhancing the value of data for participants [45].

Funded by the European Union
This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

For ACHILLES, the relevance of Common European Data Spaces is twofold. First, several project use cases operate in sensitive or regulated domains where data access, data locality, provenance, and usage restrictions are central concerns. Second, the project's longer-term sustainability and exploitation potential depend on ensuring that its outputs can remain compatible with emerging European data-sharing infrastructures and governance models. D1.1 does not implement a complete connector stack for Common European Data Spaces. Instead, it defines architectural preconditions for future compatibility. These include structured metadata, provenance and lineage information, access and usage descriptors, persistent identification of assets, integrity references, and clear documentation of provider, custodian, and validation status. These elements are necessary for future integration with data-space models because, in governed data-sharing environments, interoperability depends not only on whether data can be transmitted, but also on whether it can be discovered, interpreted, accessed, combined, and processed under explicit conditions.

The EU is currently rolling out the Common European Data Spaces across 14 sectors: agriculture, cultural heritage, energy, finance, green deal, health, language, manufacturing, media, mobility, public administration, research and innovation, skills, and tourism. The European Health Data Space [47, 48] is particularly relevant as the first domain-specific Common European Data Space and as an example of how interoperability, governance, and trusted reuse converge in a highly regulated sector. While ACHILLES is not currently presented as a platform already integrated into an operational European data space, it should remain architecturally prepared for such integration. This is especially important for use cases involving health, identity verification, and pharmaceutical compliance, where data sensitivity and accountability requirements are high.

In this sense, the contribution of Section 3 is to clarify why the interoperability layer defined in D1.1 must include governance-aware metadata, access semantics, and provenance structures. These are not optional additions to the technical architecture. They are necessary conditions for trustworthy data sharing, future data-space compatibility, and the responsible reuse of AI-related assets across the ACHILLES project. The following sections translate these legal and policy considerations into technical interoperability guidelines. Section 4 focuses on the controlled exposure of reusable capabilities through APIs, skills, plugins, and agentic interfaces. Section 5 addresses provenance, lineage, and structured documentation. Section 6 returns to European Common Data Spaces and explains how the ACHILLES interoperability layer can support future data-space compatibility.

Funded by the European Union
This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

4 API & AGENTIC CAPABILITY STANDARDISATION

This section defines the project-level interoperability expectations for exposing reusable capabilities through APIs, skills, plugins, and agentic interfaces. Instead of prescribing a single runtime or final protocol stack, it defines the minimum information that shared capabilities must expose so that they can be discovered, invoked, validated, and governed consistently within the ACHILLES IDE and the broader multi-agent environment. More detailed implementation patterns remain under the relevant technical deliverables and development tasks, especially WP3 and WP6.

This is a central concern for ACHILLES because the project integrates heterogeneous tools, datasets, models, services, and governance mechanisms across several work packages. A local tool becomes reusable at the project level only when its purpose, invocation method, inputs, outputs, dependencies, permissions, side effects, and validation requirements are explicit enough for other components, developers, and agents to use it safely. The section complements the more detailed technical work carried out in WP3 and WP6. D1.1 defines the common interoperability envelope; later implementation activities define the concrete runtime patterns, orchestration mechanisms, and IDE integration details.

For the purposes of D1.1, the following terms are used consistently:

- Tool: a callable operation with defined inputs and outputs.
- Skill: a reusable procedural capability that packages a bounded form of work together with the operational information needed for reliable reuse [6, 27, 29].
- Plugin: the packaging and integration unit through which one or more skills are exposed to the ACHILLES environment [2].
- Agent: a runtime component that manages goals, state, and the invocation of skills or tools within an execution loop [26, 39].

These definitions align the D1.1 interoperability layer with the execution architecture presented in D3.3 and with the ACHILLES IDE integration work in WP6.

4.1 Capability descriptors and plugin manifests

At the project level, ACHILLES defines a minimal standardisation baseline for reusable skills. Any skill intended for cross-work-package reuse, ACHILLES IDE integration, or agentic orchestration should be accompanied by a machine-readable and human-inspectable descriptor. At a minimum, this descriptor should include:

- identity and version;
- declared purpose and capability class;
- invocation mode;
- expected inputs and output structure;

Funded by the European Union
This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

- error semantics;
- dependencies;
- side-effect profile;
- authority and permission assumptions;
- protected-data interaction, where applicable;
- provenance-relevant outputs, where applicable;
- validation or human-review requirements, where applicable.

This descriptor constitutes the minimum capability envelope required for controlled discovery and reuse. In this framework, an API describes how an operation can be called, while a skill descriptor explains what the operation is for, under which assumptions it may be used, and what governance-relevant effects it may have.

The same principle applies to plugins. ACHILLES introduces plugin-level packaging as the standard integration surface through which reusable skills become operationally available to the ACHILLES IDE and to the shared project environment. Plugin exposure should therefore rely on explicit manifests or equivalent machine-readable packaging descriptions. These should identify the plugin, the skills it exports, the execution bindings through which those skills are made callable, the relevant runtime assumptions, dependencies, permissions, and the integration hooks needed for orchestration, provenance, and validation. Where appropriate, capability descriptors and plugin manifests should align with established interface-description approaches such as OpenAPI [25] or AsyncAPI [9], and with emerging agentic integration patterns such as MCP [22]. The choice of protocol or description format should remain proportional to the capability and its expected use. D1.1 does not impose a single technical format for all descriptors, but it does require the minimum information needed for shared interpretation, safe invocation, and later validation. At the same time, the project develops a richer and more operational notion of skill, especially where reusable capabilities combine interface contracts with procedural guidance, validation expectations, loading rules, or bounded execution behaviour [6, 29]. For this reason, T1.1 requires support for standard skill-oriented formats wherever this facilitates integration and reuse, while also enabling the ACHILLES architecture to expose richer capability structures when project needs justify them. This combination supports both interoperability with emerging standards and the technical innovation developed within the project.

4.2 Deployment and Runtime Topology

This capability-centred perspective also clarifies why standardisation in ACHILLES cannot stop at interface description alone. Once reusable skills and plugin-like capability packages enter a real runtime, the problem is no longer only whether they can be discovered and invoked, but also whether they can operate under controlled and intelligible conditions. In an agentic environment, the meaning of a capability depends not only on what it does in principle, but also on where it runs, under which identity, with what state, with what memory, and with what bounded access to the surrounding world [24]. For this reason, the deployment layer is not secondary to interoperability. It is part of

Funded by the European Union
This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

interoperability itself. At the project level, the aim should again be selective standardisation. ACHILLES does not need one universal deployment recipe for every current or future use case. It does, however, need a minimum shared discipline for those components that participate in IDE integration [2], in WP3 orchestration, or in cross-work-package workflows. The key concerns here are traceability, reproducibility, bounded execution, and operational intelligibility. These are the real reasons why packaging discipline, explicit manifests, and runtime declarations matter. They are not cosmetic engineering preferences. They are practical conditions for being able to determine what ran, under what assumptions, with which dependencies, and with which consequences.

For this reason, containerisation and sandboxing should be recommended as the default packaging discipline for deployable cross-work-package services, reusable skills that depend on stable runtime environments, and agentic components that participate in shared project infrastructure, while preserving some freedom for exploratory prototypes that have not yet crossed that threshold. At this stage, it is not considered necessary to mandate a specific technology; while the current support for Docker and Podman runtimes ensures high security, more lightweight containerisation and sandboxing alternatives such as LXC [61], Containerd [60], or Firecrackerare [59] deemed acceptable for T1.1, allowing each use case the flexibility to select the most appropriate solution for their specific development and deployment needs. The point is that once a capability becomes part of a broader multi-agent environment, it should no longer depend on ambient machine state, undocumented local assumptions, or informal credential handling. A standard containerised model, combined with explicit manifests and environment declarations, provides a sufficiently stable baseline for reproducibility without forcing the project into unnecessary operational rigidity.

Yet deployment architecture in ACHILLES is not only a question of packaging. It is also a question of topology. The term “agent” is often used too loosely, and sometimes too romantically. In practice, the ACHILLES environment may include general loop-based controllers, specialised skill executors, validators, retrieval services, sandboxed tool runners, orchestration layers, and human-supervised review points. Not every operational component should be treated as if it were a persistent persona. What matters is not anthropomorphic language, but whether each runtime entity occupies a clear role within a governed operational structure.

For this reason, D1.1 should encourage a modest but explicit deployment taxonomy around runtime identity, operational role, capability class, execution boundary, and authority scope. Identity answers which runtime entity acted. Role answers what kind of responsibility the entity was expected to fulfil. The Capability class answers whether it is primarily planning-oriented, execution-oriented, retrieval-oriented, transformation-oriented, validation-oriented, or supervisory. Execution boundary answers where it runs and with what degree of isolation. Authority scope answers what it may inspect, invoke, mutate, approve, or publish. These are not abstract distinctions for their own sake. They are the minimum vocabulary required to make a multi-agent environment inspectable rather than theatrical. The same principle applies to permissions. In conventional software systems, permission models are often discussed mainly in relation to human users. In an agentic environment, permissions must also

Funded by the European Union
This project has received funding from the European Union’s Horizon Europe research and innovation programme under Grant Agreement No 101189689.

be attached to computational actors, skill classes, and execution contexts. An entity that may inspect documents should not automatically be able to modify repositories. A component that may generate candidate changes should not automatically be able to approve or deploy them. A validator should not silently collapse into an executor. Such separations are not mere bureaucracy. They are increasingly necessary because modern AI systems are capable enough that blurred role boundaries become real operational risks [27].

A similar level of care is required for context and memory. Here, too, over-standardisation would be a mistake. ACHILLES does not require a single universal theory of memory for all agents, skills, and workflows. Different tasks require different persistence horizons, different context granularities, and different sharing patterns. Some require durable project memory. Others need only a temporary working state. Others require shared references to artefacts but not shared internal traces. D1.1 should therefore not impose a single memory model. What it should standardise are the interfaces and governance principles around memory: what may be shared, how shared context is referenced, how provenance is preserved [35], how stale state is detected, and how audit-relevant traces remain visible.

This distinction matters because conversational history is not the same thing as operational state. Intermediate reasoning traces are not the same thing as validated artefacts. Session memory is not the same thing as durable project knowledge. One of the recurring weaknesses in current industry practice is the tendency to treat all these as undifferentiated "context." A more mature agentic architecture distinguishes between durable artefacts, ephemeral working context, references to external resources, runtime state, and optional internal reasoning traces. ACHILLES should standardise enough to keep these categories visible and governable, without prematurely freezing the way each use case exploits them.

This is also where the loop-based understanding of agent systems from D3.3 becomes especially relevant [1]. Once an agent is understood not as a magical autonomous mind but as a bounded runtime repeatedly observing state, deliberating, invoking skills or tools, receiving observations, and updating working state, deployment and orchestration cease to be secondary implementation details. They become part of the core architecture. Reliability is no longer reducible to prompt wording alone. It depends on step budgets, retry rules, state compaction, traceability of tool use, explicit execution boundaries, and the disciplined management of procedural specialisation. The stronger the loop becomes as an execution pattern, the more important it becomes to make its surrounding infrastructure explicit and governable.

This does not imply that ACHILLES should reduce all coordination to deterministic orchestration. The lesson from contemporary agent engineering is more balanced. Purely conversational and weakly bounded loops are often too brittle and too opaque for higher-trust settings, but purely rigid symbolic control can easily become inflexible and expensive to maintain. The more promising direction is a hybrid architecture in which flexible model-based reasoning remains available where it adds value. At the same time, progressively stronger operational structure is imposed around it where trustworthiness depends on bounded execution, stable interfaces, durable artefacts, and

Funded by the European Union
This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

inspectable role separation. In this sense, the deployment layer should be seen as the place where the project's broader "lighter, clearer, safer" ambition acquires concrete runtime form [23].

4.3 Agent-to-Agent Communication and Choreographic Direction

This leads naturally to the question of communication between runtime entities. Once ACHILLES moves beyond isolated tool invocation and toward multi-agent orchestration, the project must consider not only how skills are exposed and deployed, but also how agents, controllers, validators, and other operational actors exchange task-relevant information. At the present stage, ACHILLES does not propose a definitive project-wide protocol for Agent-to-Agent communication across all scenarios. Such a move would be premature [18]. The landscape is still evolving, industry practice is not yet stable, and the proper degree of formalisation depends on use case, risk level, autonomy assumptions, and the desired balance between flexibility and control [36].

Still, the topic already deserves explicit architectural attention. A natural first reaction is to assume that agents can simply communicate through ordinary natural language. In many exploratory situations, that is true, and ACHILLES does not reject it. Free-form language remains highly valuable for clarification, negotiation, exploratory coordination, and the exchange of contextual knowledge. Systems inspired by patterns such as ReAct [39] and frameworks such as AutoGen [38] have shown that useful multi-step and even multi-agent behaviour can emerge from iterative language-based interaction, especially when combined with a lightweight execution framework. Language is therefore an important coordination medium, and it will remain part of the agentic landscape.

However, once the project moves from exploratory coordination toward more operationally consequential forms of orchestration, natural language alone becomes insufficient as the sole substrate. The problem is not that text is weak, but that text prioritises pragmatic flexibility rather than operational precision. A message in natural language may carry intent, explanation, and contextual nuance, yet still leave unclear what is binding, what state is being asserted, which artefacts are being referenced, whether the message is a delegation, a report, a refusal, a warning, or a final commitment, and under which role or authority the communication is taking place [36, 39]. Such ambiguity is tolerable in light coordination, but it becomes dangerous in workflows where reproducibility, auditability, bounded cost, security, and selective verification matter.

For this reason, a more plausible direction for ACHILLES is a layered communication model. Free text should remain available for explanation, negotiation, and rich contextual exchange, but it should be complemented by lighter explicit message structures for operational acts such as delegation, task acceptance, result delivery, escalation, refusal, status reporting, validation, and artefact reference. In other words, natural language should remain part of the system, but not the sole carrier of operational meaning. At the project level, the relevant standardisation target is not a universal semantic theory of all possible inter-agent conversations, but a minimum stable communication substrate—defining participants, roles, task scopes, referenced artefacts, and escalation protocols— while the D3.3 –

Funded by the European Union
This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

Multi-Agent Environment Report deliverable should provide the documentation and technical capabilities required to secure the multi-agent system and to support authorisation management, in accordance with the ACHILLES standardised framework. A related practical decision in ACHILLES concerns the interaction layer around tools and external capabilities. In connection with MCP, and as a matter of faster integration and stronger standardisation discipline, the project has converged on the view that, wherever technically reasonable, reusable capabilities should preferably also be exposed through CLI tools or CLI-oriented wrappers. This direction is discussed in greater detail in D3.3, especially in Section 2.4 [1]. The rationale is pragmatic. CLI tools can often be configured, composed, sandboxed, and invoked with very generic orchestration technology, making them easier to integrate rapidly across heterogeneous runtimes. They also tend to provide a stable operational surface that is relatively easy to inspect, script, trace, and adapt. For ACHILLES, this does not replace APIs, MCP interfaces, or richer skill packaging. Rather, it provides an additional standardisation-friendly layer that can simplify interoperability at the execution boundary and reduce friction in the practical integration of agent-usable capabilities into the ACHILLES IDE and related environments.

4.4 Protocol Landscape and Standardisation Trajectory

Task 1.1 treats Agent-to-Agent communication as a strategically important research and standardisation frontier, but not as a prematurely closed design space. A communication protocol defines the local grammar and operational structure of messages. A choreography, in the stronger sense, defines the admissible global patterns of interaction. The former concerns how agents speak. The latter concerns which larger sequences of exchange are expected, allowed, auditable, or enforceable. Once messages become sufficiently explicit, one can begin to reason not only about isolated exchanges, but also about delegation chains, approval flows, exception paths, review obligations, and security-sensitive handoffs. This is precisely the direction in which more governed multi-agent systems are likely to evolve. At the same time, D1.1 should remain cautious. ACHILLES should not yet impose one rigid choreography model across all use cases. Some interactions will remain best handled through flexible natural-language coordination. Others may require only lightweight message typing. Others, especially in higher-trust or regulated settings, may eventually justify much stronger executable choreography constraints. The present deliverable should therefore define the architectural space and the minimum substrate, rather than a final, universal closure.

The consortium should aim to write the ACHILLES specifications in a way that naturally aligns with future standardisation. This is not about creating new tasks or producing heavy RFC-style documents, but rather about using a more disciplined drafting style within our current work. By adopting the requirements classification proposed in annex A1 and using clear language with consistent terminology, we make our outputs much easier to discuss and adopt. Applying this approach to elements such as skill descriptors or interaction models ensures that our technical results are professional and ready for broader use without requiring a complete rewrite later.

Funded by the European Union
This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

4.5 Consolidated Position of the Chapter

The practical position is therefore clear. This chapter treats skills, deployment discipline, multi-agent topology, and Agent-to-Agent communication as parts of one continuous standardisation problem: how heterogeneous local capabilities become reusable skills, how such skills participate in bounded runtimes, how those runtimes are packaged and governed, and how computational actors exchange operational meaning without collapsing into either uncontrolled free text or premature over formalisation. ACHILLES is not yet defining the final shape of all skills, all runtimes, or all inter-agent protocols. It defines the common interoperability path through which these can evolve coherently at the project scale.

This path has more than one standardisation horizon. At one level, ACHILLES must standardise enough internally to support disciplined integration in the ACHILLES IDE [2] and across the broader project ecosystem. This is an ecosystem-level standardisation need imposed directly by interoperability, orchestration, and reuse requirements inside the project. At another level, the consortium should remain aligned, wherever reasonable, with emerging external standards and protocol trajectories, including those developing around APIs [9], tool exposure [25], contextual interoperability [22], and inter-agent communication [18]. Where no adequate external standard yet exists, ACHILLES may also generate open specifications, conventions, or architectural proposals that could later contribute to broader standardisation processes. In that sense, the project is not only proposing the use of existing standards but also preparing some of its open technologies and integration conventions so that they may eventually mature into reusable ecosystem proposals or, where the evidence justifies it, candidates for wider standardisation.

5 DATA PROVENANCE & LINEAGE METADATA.

ACHILLES defines provenance and lineage as central to the shared interoperability layer. The project uses the conceptual distinctions of W3C PROV [20], the event-oriented lineage perspective of OpenLineage [33], the signed-attestation logic developed in in-toto [21] and SLSA [35], and the integrity-oriented metadata discipline supported by SPDX [34]. Within the ACHILLES architecture, OpenDSU [32] is particularly relevant because it supports protected data handling alongside anchoring-based integrity and history management [30,31]. At the same time, DID-based identification [37] provides an appropriate basis for attributing provenance and for validating actions using identifiable actors and accepted verification methods. At the project level, D1.1 defines a compact provenance direction built around two complementary interoperability structures. The first is an **Asset Metadata Envelope**, which captures the current state of a dataset or derived artefact in a form suitable for discovery, policy interpretation, integrity checking, and downstream reuse. The second is a **Provenance Microledger**, which captures the ordered sequence of relevant events through which that asset reached its current state. This two-layer model gives ACHILLES a practical way to combine current-state metadata with auditable event history, while remaining aligned with the broader provenance ecosystem and with the sequential logic of AI data preparation, transformation, and validation.

Funded by the European Union
This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

This section provides a concise summary of the provenance requirements introduced in D1.1. The complete project-level formulation is specified throughout the deliverable and consolidated in **Annex A.2 Normative Requirements**. In this chapter, the role of provenance standardisation is to define the minimum interoperability conditions under which shared assets can remain traceable, verifiable, and reusable across work packages. At the same time, the detailed schemas, validation workflows, anchoring patterns, and platform exposure mechanisms are further developed through **T1.3 Data Auditing and Validation** and documented and operationalised in **WP6**.

This summary can be expressed through a small set of project-level requirements:

Table 2. Requirement IDs, classification and summaries

Requirement ID	Requirement summary
D11-PROV-01 (Mandatory)	Any dataset or derived artefact entering the shared interoperability layer shall expose a minimum metadata envelope containing persistent identity, version identity, provenance reference, integrity reference, and provider or custodian information.
D11-PROV-02 (Mandatory)	Any shared derived artefact shall preserve lineage to its declared source assets through explicit derivation references or equivalent provenance links.
D11-PROV-03 (Recommended)	Shared provenance should be represented through an append-only event-oriented structure, such as a provenance microledger, suitable for audit, validation, and downstream verification.
D11-PROV-04 (Recommended)	Provenance and validation events should be attributable to identifiable actors through accepted verification methods, including DID-based references where appropriate [37].
D11-PROV-05 (Recommended)	Shared assets should expose validation status, usage restrictions, and sensitivity information where these affect reuse, access, or training eligibility.
D11-PROV-06 (Optional)	Provenance structures may be anchored through DSU-backed or ledger-backed mechanisms where stronger notarisation, cross-organisational trust, or marketplace-oriented reuse requires it [30,32].

These requirements reflect the main architectural choices made in D1.1. First, provenance is treated as a reusable interoperability function rather than as local implementation metadata. Second, the current descriptive state of an asset is kept visible through the metadata envelope, while its transformation and validation history is preserved through an event-oriented provenance structure. Third, ACHILLES gives explicit importance to cryptographic and organisational trust. A provenance statement is therefore more useful when it is linked to verifiable identity through DIDs [37], when the integrity of the relevant artefact is preserved through hashes or equivalent mechanisms, and when anchoring can provide stronger notarisation support through OpenDSU-style structures [30,32].

Funded by the European Union
This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

This model also supports a practical progression path for the project. The metadata envelope provides the minimum descriptive surface needed by tools, plugins, and agents for discovery and controlled reuse. The provenance microledger provides the historical and evidentiary surface needed for audit, validation, and later policy enforcement. Together, they support a shared interoperability layer in which assets can be consumed not only as files or payloads, but also as traceable, governance-aware project artefacts.

From this perspective, the value of provenance standardisation in ACHILLES is straightforward. It enables shared datasets and derived artefacts to remain inspectable across transformations, strengthens the basis for trustworthy training and fine-tuning decisions, and supports future integration with validation services, marketplace functions, and policy-aware data-sharing workflows. The detailed implementation and documentation of these structures are therefore naturally carried forward in T1.3 and WP6, where the project turns these high-level interoperability requirements into concrete schemas, services, and integration mechanisms.

5.1 Structured AI Documentation as a Provenance-aware Interoperability Layer

The provenance and metadata structures proposed in this chapter address the traceability and integrity of datasets and derived artefacts as they move through the ACHILLES pipeline. A closely related but distinct challenge is documenting the AI components themselves: the models, datasets, AI systems, risk assessments, and compliance evidence that collectively define the governance state of a project's outputs. In practice, AI documentation is often fragmented across spreadsheets, slide decks, wikis, and manually assembled reports, with no structured connection to the provenance records, validation events, or approval workflows that give this documentation its evidentiary value. When documentation and provenance evolve independently, the result is duplicated information, version drift, and governance artefacts that are difficult to audit, reuse, or project to different stakeholder audiences.

To address this gap within ACHILLES, the Cardsmith framework is being developed (in WP5, T5.4) as a structured AI documentation system designed to complement the provenance and metadata layer defined in this deliverable. Cardsmith manages documentation through canonical card types, including ModelCard, DataCard, AISystemCard, RiskAssessmentCard, and ChecklistCard, stored in a shared schema with JSON-LD context and ontology-backed field mappings. This creates a consistent, machine-readable representation for technical, risk, and compliance information that can be validated, versioned, and projected into stakeholder-specific views without duplication.

The relevance of Cardsmith to D1.1 lies in three areas. First, Cardsmith extends the provenance principles of this chapter to the documentation layer itself. Every field update in a Cardsmith card records provenance metadata: who generated it, where it came from, how it was created, and its current approval status. Automated agents and connectors, such as metadata importers from MLflow or Hugging Face, typical developer-centric tools, can draft content, but approval-sensitive fields remain subject to human review. This approach to provenance at the field level is architecturally consistent with the append-only Provenance Microledger model proposed earlier in this section: both share the

Funded by the European Union
This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

principle that the history of an artefact's evolution must be preserved, attributable, and inspectable, not merely its final state. In this sense, Cardsmith treats AI documentation not as a static deliverable but as a governance artefact with provenance as a first-class concept.

Second, Cardsmith contributes a semantic interoperability layer through its use of established ontologies. The framework maps card fields to concepts from AIRO (AI Risk Ontology) [49], MCRO (Machine Learning Card Reporting Ontology) [50], DPV (Data Privacy Vocabulary) [52], and the AIDOC-AP (AI Documentation Application Profile) [56], providing an ontological mapping specifically aligned with the EU AI Act Annex IV [51] documentation requirements. This ontological grounding enables documentation artefacts to be not only structurally consistent but also semantically interoperable with external governance vocabularies. For ACHILLES, this is a significant complement to the JSON-based provenance profiles defined earlier in this section. While the Asset Metadata Envelope [33] and Provenance Microledger [35] ensure structural and cryptographic traceability of data assets, the ontology-backed documentation layer ensures that the meaning and regulatory relevance of those assets can be communicated across organisational and jurisdictional boundaries. The combination of both layers, structural provenance and semantic documentation, strengthens the project's interoperability posture in ways that neither layer achieves alone [54].

Third, Cardsmith supports audience-specific projection and compliance workflows that are directly relevant to the European regulatory context discussed in Section 6 [47]. The same canonical card can be projected into views tailored for developers, auditors, operators, affected subjects, regulators, and public audiences, with outputs rendered in different formats (e.g., HTML, PDF, and RDF) [55]. Built-in checklist templates support Fundamental Rights Impact Assessment (FRIA) [57], Data Protection Impact Assessment (DPIA) [58], EU AI Act Annex IV documentation [51], and related governance workflows. For ACHILLES, this means that the documentation generated during development and validation activities can be reused across internal review processes, external audit requirements, and future data-space participation, without requiring separate documentation efforts for each audience or regulatory context.

From an integration perspective, Cardsmith operates as a library-first Python framework that can be embedded into existing AI governance workflows or exposed as an internal API service. It does not impose a separate infrastructure requirement on the ACHILLES architecture. Instead, it is designed to consume metadata from the provenance [35] and lineage [33] structures proposed in this chapter, including asset identifiers, derivation links, validation status, and sensitivity classifications, and to enrich them with structured documentation, ontological context, and stakeholder-specific presentation. Conversely, documentation artefacts managed by Cardsmith can themselves be referenced from the Asset Metadata Envelope, creating a bidirectional link between provenance-carrying data assets and their associated governance documentation.

The relationship between Cardsmith and other project activities is specified. Within WP5, Task 5.4 and Deliverable D5.3 (Data/Model Cards Report, M38) address improved and automated methods for generating model and data cards. Cardsmith provides the underlying framework and tooling

Funded by the European Union
This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

infrastructure on which these automated generation methods can be built, validated, and operationalised. Within WP4, the legal and ethical mapping work and the ethics guidelines deliverables define the regulatory and ethical requirements that Cardsmith's compliance templates and checklist workflows are designed to support [53]. Within WP6, the ACHILLES IDE can surface Cardsmith's documentation and validation capabilities as integrated governance features, consistent with the plugin and capability exposure principles discussed in Section 4 of this deliverable. In this sense, Cardsmith is not a standalone documentation tool but a cross-cutting enabler that connects the provenance substrate of D1.1 to the compliance, transparency, and reporting requirements distributed across multiple work packages [54].

In summary, the proposed standardisation direction is the following. ACHILLES should adopt a small JSON specification with two linked layers: an Asset Metadata Envelope for the current state and policy, and a Provenance Microledger for event-level history [54]. The microledger should explicitly encode actions of modification and validation, as well as the hashes necessary to verify each transition. The model should support DID-based attribution, trust-root references, and optional anchoring through DSUs or as independent microledgers. Complementing these structural and cryptographic layers, the Cardsmith framework extends the same provenance-aware principles to AI documentation artefacts, providing ontology-backed card types, field-level provenance, validation pipelines, and audience-specific projections grounded in vocabularies such as AIRO [49], DPV [52], and AIDOC-AP [56]. Together, the provenance infrastructure and the structured documentation layer provide the project with a realistic, technically grounded path toward trustworthy data sharing, dataset validation, agent-ready cryptographic provenance, and governance-ready AI documentation that can support internal review, regulatory reporting, and future data-space participation [53].

Funded by the European Union
This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

6 INTEGRATION WITH EUROPEAN COMMON DATA SPACES (ECDS)

The alignment of ACHILLES with the European Common Data Spaces [11] is relevant at both strategic and architectural levels, particularly for regulated sectors such as health and pharmaceuticals. ECDS interoperability is not reduced to data exchange: operational data spaces rely on a concrete stack of building blocks, including DCAT-AP [55] for dataset description, ODRL [52] for usage policy expression, Gaia-X self-descriptions [16] for participant and service trust, and the IDSA Dataspace Protocol for connector-level interactions, consolidated into the DSSC Blueprint [10]. For ACHILLES, and in particular for the HERA pharmaceutical use case, this means that interoperability must address data sovereignty, policy-constrained access, provenance, accountability, and secure cross-organisational collaboration as first-class concerns rather than as optional add-ons. Figure 1 illustrates the EHDS architecture and its interoperability layers as they relate to the ACHILLES project context.

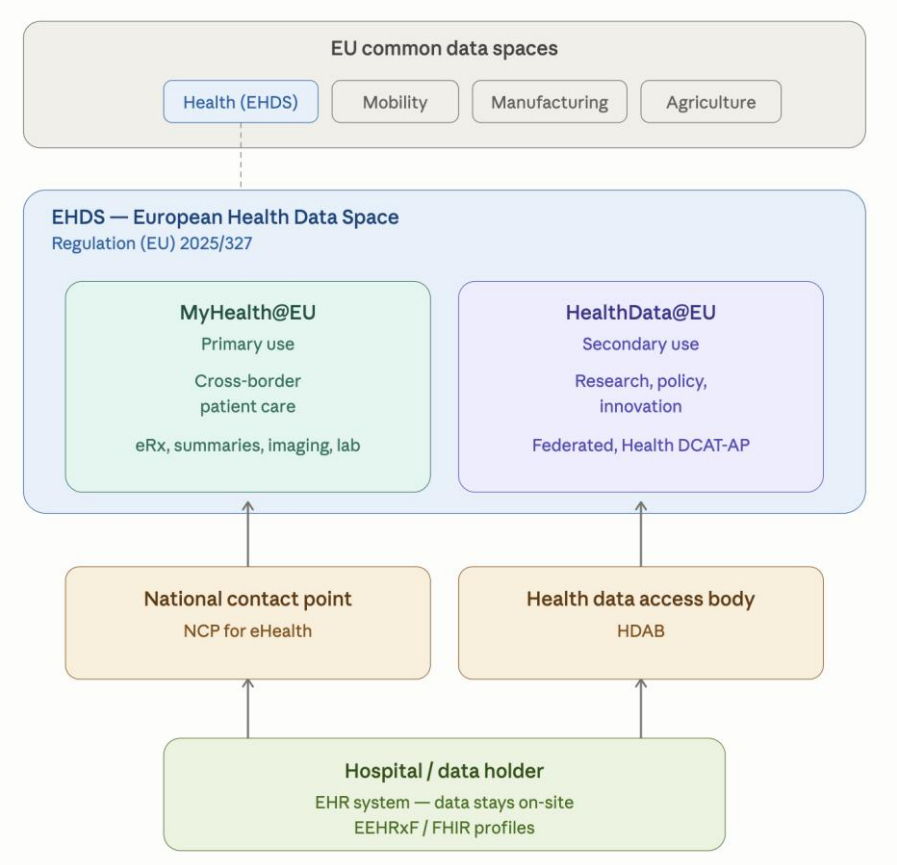

Figure 1 - EHDS architecture and interoperability layers relevant to ACHILLES

At the current stage, ACHILLES is not being presented as a platform already integrated into an operational ECDS. Since the DSSC Blueprint [10] and EHDS secondary acts are still evolving, D1.1 defines architectural preconditions for future compatibility rather than a concrete connector implementation. The objective of this chapter is therefore to describe the project's current

Funded by the European Union
This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

understanding of the ECDS model, to explain its relevance for ACHILLES, and to define those standardisation choices that are useful already now, even before full operational data-space integration becomes feasible. This perspective is reinforced by the evolution of the European Health Data Space. EHDS (Regulation (EU) 2025/327 [47]) entered into force on 26 March 2025, with primary-use obligations applying from March 2029 and secondary-use obligations phased through 2031, supported by implementing and delegated acts.

For ACHILLES, this staged operationalisation means that the most appropriate response is not premature technical closure around assumptions that may later change, but architectural preparedness. The project can already benefit from adopting design principles that are consistent with the direction of EHDS and the broader European data strategy: structured metadata, explicit provenance, policy-aware access models, trustworthy identity layers, and execution patterns that respect data locality and institutional control.

Within this context, the most relevant contribution of the ECDS model is that it reframes interoperability as a governed capability rather than as a purely syntactic property. A resource is not interoperable merely because it can be serialised, transmitted, and parsed. In regulated environments, interoperability also depends on whether the resource can be discovered, interpreted, accessed, combined, and processed under explicit and enforceable conditions. These conditions include legal entitlement, organisational role, declared purpose [12], security posture, traceability obligations [54], and restrictions on onward use [52]. In ECDS practice, such conditions are increasingly expressed through ODRL [52], the W3C-standard policy language adopted as the common vocabulary for usage control across European data spaces. This broader understanding is highly compatible with the general position adopted in D1.1, namely that modern interoperability must combine technical compatibility with explicit governance and machine-readable contextualisation.

For ACHILLES, this has immediate consequences for internal standardisation. The project does not need to define a complete project-wide ontology for all future ECDS interactions. It does, however, need a stable descriptive envelope around datasets, models, services, plugins, and derived artefacts. The Asset Metadata Envelope defined in Section 5 already provides most of the required fields: persistent identification, modality, provenance reference, integrity, provider or custodian identity, validation status, sensitivity classification, and usage restrictions. These requirements are not external additions to the ACHILLES architecture; they are direct extensions of the provenance and interoperability principles already established in Chapters 3 to 5.

The first area in which this alignment becomes technically concrete is metadata. European data spaces depend on discoverability and machine-processable resource descriptions [11]. For ACHILLES, this implies that internal data structures should not remain narrowly tied to local execution needs. They should support a richer descriptive layer that can be projected into DCAT-AP for external discoverability, In practical terms, this means that tabular data, textual corpora, images, model artefacts, and service outputs should all be accompanied by metadata that can survive ingestion,

Funded by the European Union
This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

transformation, orchestration, and reuse. Without this layer, later interoperability with external data spaces would remain largely superficial.

A second area is trust and identity. In European data spaces, participation is not modelled as anonymous data exchange. It is structured around identifiable participants, declared roles, verifiable claims, and trust frameworks that allow actors to assess one another under shared rules [15] such as Gaia-X Trust Framework principles [16], verifiable self-description mechanisms, and architecture-level governance constructs [17], with the IDSA Dataspace Protocol and the DSSC Blueprint [10] providing complementary connector-level and operational guidance. This is one of the reasons why GAIA-X remains relevant for ACHILLES. Its importance does not lie in the assumption that every project component must immediately implement the full GAIA-X stack. Rather, its value lies in the conceptual and technical vocabulary it provides for self-description, machine-readable governance, trust assertions, and policy-aware service exposure. These concepts are directly relevant to a platform such as ACHILLES, where future interoperability may involve multiple organisations, heterogeneous infrastructures, and differentiated rights over data and computation.

In this regard, the notion of self-description is particularly useful. A connector, service, or plugin intended to interact with external sovereign data environments should expose not only interface-level information, but also organisational and policy-relevant metadata. This may include the operator identity, compliance posture, type of resource exposed, admissible modes of access, security assumptions, and restrictions on downstream processing. This aligns with the capability descriptor model of Section 4, which should be designed so that it can later be projected into a Gaia-X-compatible self-description. Within D1.1, the role of this principle is preparatory. It does not require the project to define a complete conformity regime, but it does justify the inclusion of metadata structures that can later be mapped to European trust and policy frameworks.

A third area concerns access control and policy expression. In regulated data environments, the distinction between discoverability and accessibility is essential. A dataset may be visible in a catalogue without being exportable. A resource may support approved computation without allowing raw download. A model or service may expose derived results while restricting access to the underlying inputs. These distinctions are central to the governance logic of European data spaces, especially in health-related settings [44], and correspond to the policy patterns expressed in ODRL [52]. ACHILLES should therefore avoid flat notions of availability. Instead, the architecture should progressively support differentiated access semantics, including metadata-only discovery, controlled retrieval, local-only execution, federation participation, and restricted result exchange. This also implies a tighter integration among access control lists, usage policies, and provenance records, so that the conditions of access remain visible across subsequent processing stages.

A fourth area is semantic interoperability across heterogeneous modalities. The data relevant to ACHILLES use cases is unlikely to be homogeneous. It may include structured tables, text collections, image data, reports, annotations, model artefacts, and workflow-generated derivatives. The challenge is therefore not merely format conversion but maintaining sufficient coherence across

Funded by the European Union
This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

these modalities to support reliable reuse, validation, and regulated collaboration. The European data space agenda does not eliminate this heterogeneity; rather, it requires infrastructures to handle it in a disciplined way. For ACHILLES, the appropriate response is a layered approach. At the lower level, the project should define stable wrappers for core modalities. At the middle level, these wrappers should carry provenance, schema references, and policy-relevant metadata. At a higher level, semantic mediation may be assisted by agents or language models, but this should remain bounded by explicit contracts, review points, and domain-specific validation whenever the context is high-risk. The Cardsmith framework, discussed in Chapter 5, illustrates this layered approach for the specific case of AI documentation: its ontology-backed card types map documentation fields to established governance vocabularies such as AIRO, DPV, and AIDOC-AP, providing a concrete semantic interoperability layer that complements the structural provenance model without requiring a single project-wide ontology. This position remains consistent with the broader argument of D1.1 that flexible AI-based mediation is useful, but not sufficient on its own.

The execution dimension is even more important. In many ECDS-relevant settings, the primary issue is not whether data can be moved, but whether useful computation can be performed without violating local control constraints. This is why the alignment between ACHILLES and European data spaces naturally leads to federated, privacy-preserving computational models. Data locality, constrained sharing, controlled access, and institutionally bounded processing are all more compatible with federated learning [11], secure aggregation [13], secure multi-party computation, confidential execution environments, and related approaches than with centralised data aggregation. For this reason, the most substantial technical continuation of the present chapter lies in T2.3 Federated Learning and Secure Computation Strategies.

The division of labour between D1.1 and T2.3 should therefore be stated clearly. D1.1 does not define the final protocol stack for federated learning, secure aggregation, privacy-preserving analytics, or confidential computation. These topics depend on the selected framework, use-case requirements, performance constraints, and the technical findings of WP2. What D1.1 can provide is the interoperability substrate on which such work can later rely. This includes standardised identifiers for resources and transformations, provenance-carrying metadata, explicit access and execution modes, modular policy expression, and architectural separation between discovery, access, computation, and result dissemination. Without these elements, later secure computation work would remain disconnected from the broader interoperability model of the project.

In this sense, the present chapter primarily contributes at the level of the architectural discipline. It defines the conditions under which future ECDS-oriented integration becomes technically plausible and institutionally credible. It also explains why certain forms of standardisation are justified already now. Standardising provenance, access semantics, data descriptors, and connector metadata is not a peripheral exercise. It is a necessary step toward any future form of sovereign machine learning in distributed, regulated data environments. Conversely, standardising the internal payloads of specific

Funded by the European Union
This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

federated learning frameworks too early would be less appropriate in D1.1, since those decisions depend on later technical choices that should remain under WP2 control.

This distinction is also important from a strategic point of view. The relevance of ECDS to ACHILLES extends beyond immediate project execution. It also concerns the long-term usability of project results beyond the funded period. If the internal data ecosystem of ACHILLES evolves in a way that aligns with European trust, governance, and interoperability standards, then the project outputs may remain useful in future data-sharing infrastructures, regulated research environments, and sovereign AI workflows. This is particularly relevant for pharmaceutical and health-related use cases, where interoperability increasingly functions not only as a technical convenience, but also as a market, regulatory, and institutional requirement.

For the HERA-oriented use case, this longer-term relevance is substantial. Pharmaceutical and health innovation depends on the ability to operate across fragmented institutional landscapes, with strong asymmetries in data stewardship, permissions, and evidentiary practices. A platform that combines multi-source ingestion, explicit provenance, governed skill exposure, and future support for federated or secure computation is better positioned for such environments than one built around simple data aggregation. In this respect, ECDS alignment strengthens both the policy relevance and the potential exploitation value of ACHILLES. It allows the project to frame its interoperability work not as generic infrastructure, but as preparation for participation in the next generation of European data-intensive ecosystems.

The resulting position is therefore limited, but concrete. D1.1 introduces the conceptual and architectural basis for future compatibility with European Common Data Spaces. It does so by treating provenance, metadata, trust descriptors, access policies, and multi-modal interoperability as project-level concerns. It does not yet attempt to fix the final secure computation mechanisms through which these principles will later be operationalised. That effort belongs primarily to T2.3. The link between the two is straightforward: D1.1 defines the interoperability structure; T2.3 will define the computation strategies that exploit this structure under conditions of sovereignty, privacy, and distributed control.

In conclusion, the main contribution of this chapter is to position ECDS alignment as a concrete interoperability requirement for ACHILLES, rather than as an external policy reference. Its relevance lies in the way it clarifies the architectural implications of sovereign data sharing: richer metadata (DCAT-AP-compatible), stronger provenance, explicit policy layers ODRL-mappable), machine-readable trust descriptors, and support for locality-preserving computation. These are already useful within the project and provide a credible basis for future integration with European data ecosystems (as outlined in requirements D11-ECDS-01 to D11-ECDS-03 in Annex A) as these become more mature and operational.

Funded by the European Union
This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

7 CONCLUSIONS

D1.1 defines the project-level interoperability foundation for ACHILLES. It establishes the common structures needed to support consistent integration, reuse, validation, and governance of data assets, AI artefacts, services, plugins, skills, and agent-usable capabilities across the project. The deliverable adopts a selective and pragmatic standardisation approach. It does not attempt to define every domain ontology, runtime protocol, federated-learning payload, secure computation mechanism, or European Data Space connector stack. Instead, it focuses on the cross-cutting structures that are needed early in the project to reduce fragmentation and support coherent cross-work-package development. The main contribution of D1.1 is the definition of a minimum interoperability layer comprising:

- capability descriptors and plugin manifests for reusable APIs, skills, services, and agent-facing capabilities;
- runtime, permission, access, side-effect, and validation declarations for shared components;
- metadata, provenance, lineage, and integrity structures for datasets and derived artefacts;
- structured AI documentation mechanisms, including model cards, data cards, system cards, risk assessments, and compliance checklists;
- architectural assumptions for future alignment with European Common Data Spaces.

These elements provide the basis for making ACHILLES components both machine-usable and human-inspectable. They support discovery, controlled invocation, traceability, documentation, validation, and governance-aware reuse. This is particularly important for the ACHILLES IDE and multi-agent environment, where tools and data assets must be exposed not only as technical endpoints or files, but as capabilities and artefacts with explicit purpose, assumptions, constraints, provenance, and validation evidence. D1.1 also clarifies the boundary between project-level interoperability and work-package-specific implementation. WP1 defines the common interoperability expectations. WP2, WP3, WP6, and WP7 will refine, implement, and validate these expectations through privacy-preserving computation, multi-agent orchestration, IDE integration, and use-case evaluation. This preserves the flexibility needed for research and engineering while ensuring that components entering the shared ACHILLES environment follow a common interoperability discipline.

The guidelines introduced in this deliverable should therefore be applied when a component, dataset, model, service, plugin, agent, connector, or documentation artefact is intended for cross-work-package reuse, ACHILLES IDE integration, regulated data-sharing workflows, or common validation. Internal prototypes may continue to evolve locally until they enter this shared interoperability layer. Overall, D1.1 provides a stable but adaptable foundation for the next phases of ACHILLES. It supports technical integration, strengthens traceability and governance, prepares the project for future data-space compatibility, and contributes to the broader objective of developing AI systems that are lighter, clearer, and safer.

Funded by the European Union
This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

8 ANNEX A. INTEROPERABILITY REQUIREMENTS AND COMPLIANCE MATRIX

This annex consolidates the project-level interoperability requirements defined throughout D1.1 into a compact normative form. Its purpose is not to introduce a new architecture, but to make explicit which interoperability obligations apply across ACHILLES, who is expected to implement them, and how compliance can later be validated.

The annex should be read together with Chapters 3 to 6. Where the main body explains the rationale and architectural boundaries, the present annex provides a more operational summary of the common requirements that support cross-work-package integration, trustworthy capability exposure, provenance-aware data reuse, and future compatibility with European data-space environments.

The requirements below apply only to components, datasets, services, plugins, agents, or connectors that are intended for cross-work-package reuse, ACHILLES IDE integration, regulated data-sharing workflows, or participation in the common interoperability layer. Internal exploratory prototypes that remain confined to a local task or work package are not required to comply fully until they cross into the shared project environment.

A.1 Requirement Classification

For the purposes of this annex, requirements are classified as follows.

Mandatory requirements must be satisfied by any component entering the project's shared interoperability layer.

Recommended requirements are strongly encouraged because they improve traceability, safety, or future integration potential, but may depend on implementation maturity or use-case needs.

Optional requirements are useful where relevant, but are not required project-wide at this stage.

Specific requirements in Annex A.2 were classified based on implementation team feedback and exploitation goals. To ensure the system supports the intended use cases while remaining flexible for highly regulated enterprise clients, some aspects will be negotiated directly. We have prioritised implementation efficiency here, as these points often allow for multiple solution paths.

A.2 Normative Requirements

Table 3 - Interoperability Normative Requirements

Requirement ID	Requirement
D11-API-01 (Mandatory)	Any ACHILLES capability intended for cross-WP reuse shall expose a machine-readable capability descriptor.
D11-API-02 (Mandatory)	Capability descriptors shall declare at minimum: capability identifier, purpose, input expectations, output structure, invocation mode, and error semantics.

Funded by the European Union
This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

D11-API-03 (Mandatory)	Capabilities with state-changing behaviour shall explicitly declare whether they are read-only, transformative, or governance-sensitive.
D11-API-04 (Recommended)	Capabilities exposed through synchronous HTTP interfaces should use OpenAPI-compatible descriptions; event-driven interfaces should use AsyncAPI-compatible descriptions where applicable.
D11-API-05 (Recommended)	Where a capability is intended for agent-facing tool use, its descriptor should be compatible with MCP-style exposure or be mappable to such exposure through a wrapper.
D11-PROV-01 (Mandatory)	Any dataset or derived artefact intended for cross-WP use shall carry a minimum metadata envelope containing persistent identity, version identity, provider or custodian identity, modality, format, integrity reference, and provenance reference.
D11-PROV-02 (Recommended)	Any derived dataset used in shared workflows should record derivation links to its source assets through a derived from or equivalent reference.
D11-PROV-03 (Mandatory)	Metadata envelopes shall include at least one integrity mechanism, such as a content hash, to support later verification.
D11-PROV-04 (Recommended)	Assets intended for trustworthy reuse should expose validation status, usage restrictions, and sensitivity class, where applicable.
D11-PROV-05 (Recommended)	Provenance histories should be represented as append-only event sequences, or be mappable to such a structure, for assets whose transformation history is relevant to auditing, validation, or regulated reuse.
D11-EXEC-01 (Mandatory)	Any deployable agentic component entering the shared environment shall declare runtime identity, operational role, execution boundary, and authority scope.
D11-EXEC-02 (Mandatory)	Agentic components shall not implicitly inherit permissions beyond those explicitly assigned by their deployment and orchestration context.
D11-EXEC-03 (Recommended)	Shared workflows should preserve references to the artefacts, datasets, or external resources used during execution, so that downstream inspection and audit remain possible.
D11-EXEC-04 (Recommended)	Components intended for shared deployment should use containerised or equivalently reproducible packaging unless a justified exception is documented.
D11-ECDS-01 (Mandatory)	Assets, services, and connectors expected to participate in future sovereign or regulated data-sharing environments shall expose metadata sufficient to support discoverability, provenance, access distinctions, and steward or provider identification.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

D11-ECDS-02 (Mandatory)	The shared interoperability layer shall distinguish between discovery, access, local execution, federated participation, and restricted result exchange wherever these distinctions are relevant to the resource.
D11-VAL-01 (Mandatory)	Compliance with the present requirements shall be validated through a combination of schema validation, integration testing, implementation review, and use-case validation in the relevant work packages.

A.3 Compliance Matrix: Guidelines for Work Packages

Table 4 - Guidelines and Evidence

Requirement ID	Guidance
D11-API-01	Applies to WP3 and WP6 integration surfaces, as well as any reusable service or plugin exposed to the IDE or shared orchestration layer. Compliance evidence may include OpenAPI, AsyncAPI, MCP-compatible descriptors, or equivalent machine-readable manifests.
D11-API-02	The minimum descriptor is validated through a schema-level or checklist-based review. This requirement ensures that capabilities are not exposed only through informal documentation.
D11-API-03	Validation may be performed through manifest review and integration tests that verify that the side-effect class and operational expectations are explicit.
D11-API-04	This is primarily a recommended conformance direction for the interoperability discipline. Exceptions may apply to immature research prototypes that have not yet been shared.
D11-API-05	This is particularly relevant for WP6 and ACHILLES IDE integration. A native MCP implementation is not mandatory if equivalent agent-facing exposure can be produced through wrappers.
D11-PROV-01	Applies to datasets and derived artefacts used across work packages, especially where trust, explainability, validation, or regulated reuse matter. Compliance evidence may include metadata JSON files or registry entries.
D11-PROV-02	This supports lineage and auditability. Validation may be performed by checking whether shared assets include backward references to their immediate or declared source artefacts.
D11-PROV-03	At least one stable hash-based integrity reference should be present. Stronger cryptographic attestations may be added where relevant.

Funded by the European Union
This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

D11-PROV-04	This requirement is especially relevant for validation-sensitive or regulated assets. Evidence may include metadata fields for validation status, usage restrictions, and sensitivity labels.
D11-PROV-05	This is particularly relevant to T1.3 and to any provenance-aware data-sharing functions surfaced later in WP6. Event sequences may be implemented as microledgers or equivalent append-only provenance structures.
D11-EXEC-01	Applies to reusable agents, orchestration participants, validation services, and governed execution components. Runtime manifests, deployment descriptors, or registry entries may provide compliance evidence.
D11-EXEC-02	This requirement is validated through deployment policy review and role or permission separation in the orchestration environment.
D11-EXEC-03	Shared executions should leave enough trace to reconstruct which artefacts or external resources influenced downstream outputs.
D11-EXEC-04	This requirement improves reproducibility and reduces hidden assumptions about the environment. Exceptions should be documented where containerisation is inappropriate or premature.
D11-ECDS-01	Applies to future-facing connectors, datasets, services, and artefacts likely to interact with regulated or sovereign data-sharing settings. Compliance may be partial at the current project stage, but metadata preparedness should be visible.
D11-ECDS-02	This requirement ensures that discoverability is not conflated with unrestricted access or unrestricted data movement. Validation may be checklist-based until more concrete connector implementations exist.
D11-ECDS-03	This requirement is a scope boundary. Its purpose is to prevent D1.1 from being misread as the implementation owner for framework-specific secure computation or federated execution stacks.
D11-VAL-01	Validation responsibility is distributed: WP6 for capability exposure and IDE integration; WP3 for multi-agent execution assumptions; WP2 for secure and federated computation compatibility; and WP7 for validation in concrete use cases.

A.4 Implementation Ownership

The requirements in this annex do not imply that all implementation work belongs to WP1. Their role is to define the common interoperability layer on which later implementation work can rely.

WP2 is expected to implement the secure and federated computational mechanisms that consume the identifiers, metadata structures, and access distinctions defined here.

WP3 is expected to implement the orchestration logic, execution boundaries, and agentic

Funded by the European Union
This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

runtime structures that operationalise the relevant capability and execution requirements.

WP6 is expected to materialise many of these requirements in the ACHILLES IDE through plugin manifests, capability registries, discovery surfaces, and provenance-aware integration patterns.

WP7 is expected to validate the adequacy of these structures in concrete use-case settings.

A.5 Out-of-Scope Clarification

The present annex does not impose a full project-wide ontology for domain semantics, a finalised Agent-to-Agent communication protocol, a universal memory model, a framework-specific federated learning payload format, a secure computation protocol or a complete compliance stack for European Common Data Spaces. These remain either use-case-dependent or under the responsibility of later technical work packages. This annex provides a concise normative summary of the minimum interoperability expectations that support shared integration across ACHILLES.

A.6 Minimum Validation Evidence

To ensure that the interoperability requirements defined in this annex remain operational rather than merely declarative, each shared ACHILLES component should be accompanied by a minimum body of validation evidence appropriate to its role. The purpose of this evidence is not to create unnecessary bureaucracy, but to make compliance inspectable across work packages and to support later integration, audit, and use-case validation.

The precise evidence required may vary depending on whether the object under review is a dataset, a plugin, an agentic component, a connector, or a shared service. Nevertheless, a common minimum set of evidence types can already be identified at the project level.

Table 5 - Minimum Validation Evidence for ACHILLES Components

Evidence Type	Description
Capability Descriptor	A machine-readable description of the exposed capability, such as an OpenAPI document, AsyncAPI document, MCP-compatible tool descriptor, or equivalent manifest showing inputs, outputs, invocation mode, and error semantics.
Plugin or Service Manifest	A structured manifest identifying the component, its version, operational purpose, dependencies, side-effect class, and integration assumptions within the ACHILLES environment.
Asset Metadata Envelope	A metadata record for any shared dataset or derived artefact, containing the minimum fields required for identity, provenance reference, integrity, provider or custodian information, and usage-related restrictions where applicable.

Funded by the European Union
This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

Provenance Record	A provenance microledger, append-only event log, or equivalent structured lineage record showing the relevant transformations, validations, or approval events associated with a shared asset.
Integrity Evidence	At least one verifiable integrity reference, such as a content hash, signed metadata object, or equivalent cryptographic attestation sufficient to support later verification of the shared artefact.
Deployment Descriptor	A container manifest, reproducible runtime specification, or equivalent deployment description showing how the component is packaged and under which execution assumptions it operates.
Identity and Role Declaration	A record identifying the runtime component, its declared operational role, execution boundary, and authority scope, especially for reusable agentic services or orchestration participants.
Access and Policy Declaration	A structured declaration of the intended access mode or policy posture of the resource, including relevant distinctions such as discovery-only, controlled access, local execution, federated participation, or restricted result exchange.
Integration Test Evidence	Test logs, validation reports, or demonstrator outputs showing that the component can be discovered, invoked, or consumed correctly through the shared interoperability layer.
Use-Case Validation Evidence	Evidence produced in WP7 or other concrete validation settings showing that the proposed interoperability structures remain adequate in practical scenarios rather than only in abstract design.

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

9 REFERENCES

[1] ACHILLES Project, 2026. Deliverable D3.3 – Multi-Agent Environment Report (v1). Internal deliverable.

[2] ACHILLES Project, 2026. Deliverable D6.1 – ACHILLES IDE (v1). Internal deliverable.

[3] Alboaie, L., Alboaie, S., Barbu, T., 2014. Extending Swarm Communication to Unify Choreography and Long-Lived Processes, https://www.researchgate.net/publication/265381241_Extending_swarm_communication_to_unify_choreography_and_long-lived_processes

[4] Alboaie, S., Alboaie, L., Bogdan, I., Vaida, M., 2023. Bringing Executable Choreographies to the IoT, https://www.researchgate.net/publication/372769620_Bringing_executable_choreographies_to_the_IoT

[5] Anthropic, 2024. Building Effective AI Agents, https://www.anthropic.com/research/building-effective-agents

[6] Anthropic, 2025. Agent Skills / Tool Use and Capability Design Guidance, https://platform.claude.com/docs/en/agents-and-tools/agent-skills/overview

[7] Anthropic, 2025. Context Windows & Context Management Documentation, https://platform.claude.com/docs/en/build-with-claude/context-windows

[8] Anthropic, 2025. Tool Use / Tools Documentation, https://platform.claude.com/docs/en/agents-and-tools/tool-use/implement-tool-use

[9] AsyncAPI Initiative, 2026. AsyncAPI Specification 3.1.0 / Latest Specification Documentation, https://www.asyncapi.com/docs/reference/specification/latest

[10] Data Spaces Support Centre, 2026. Data Spaces Blueprint, https://blueprint.dssc.eu/

[11] European Commission, 2026. Common European Data Spaces, https://digital-strategy.ec.europa.eu/en/policies/data-spaces

[12] European Union, 2022. Regulation (EU) 2022/868 on European data governance (Data Governance Act), https://eur-lex.europa.eu/eli/reg/2022/868/oj/eng

[13] European Union, 2023. Regulation (EU) 2023/2854 on harmonised rules on fair access to and use of data (Data Act), https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32023R2854

[14] European Union, 2025. Regulation (EU) 2025/327 of the European Parliament and of the Council on the European Health Data Space, https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=OJ:L_202500327

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

[15] Gaia-X, 2022. Gaia-X Trust Framework, https://docs.gaia-x.eu/policy-rules-committee/trust-framework/22.10/

[16] Gaia-X, 2022. Self-Description Definition, Gaia-X Architecture Document, https://docs.gaia-x.eu/technical-committee/architecture-document/22.10/self-description/

[17] Gaia-X, 2025. Gaia-X Architecture Document, 25.05 Release, https://docs.gaia-x.eu/technical-committee/architecture-document/latest/changelog/

[18] Google Developers Blog, 2025. Announcing the Agent2Agent Protocol (A2A), https://developers.googleblog.com/en/a2a-a-new-era-of-agent-interoperability/

[19] Google Developers Blog, 2025. Google Cloud donates A2A to Linux Foundation, https://developers.googleblog.com/en/google-cloud-donates-a2a-to-linux-foundation/

[20] Groth, P., Moreau, L., et al., 2013. PROV-Overview: An Overview of the PROV Family of Documents. W3C, https://www.w3.org/TR/2013/NOTE-prov-overview-20130430/

[21] in-toto, 2025. Getting started / layout and link metadata, https://in-toto.io/docs/getting-started/

[22] Model Context Protocol, 2025. Specification, https://modelcontextprotocol.io/specification/2025-11-25

[23] NIST, 2023. AI Risk Management Framework (AI RMF 1.0), https://www.nist.gov/itl/ai-risk-management-framework

[24] NIST, 2024. Artificial Intelligence Risk Management Framework: Generative Artificial Intelligence Profile, https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.600-1.pdf

[25] OpenAPI Initiative, 2025. OpenAPI Specification v3.2.0, https://spec.openapis.org/oas/v3.2.0.html

[26] OpenAI, 2025. A Practical Guide to Building Agents, https://cdn.openai.com/business-guides-and-resources/a-practical-guide-to-building-agents.pdf

[27] OpenAI, 2025. Agents Overview / Cookbook Topic: Agents, https://developers.openai.com/cookbook/topic/agents/

[28] OpenAI, 2025. MCP and Connectors, https://developers.openai.com/api/docs/guides/tools-connectors-mcp/

[29] OpenAI, 2026. Skills / Tool Use and Capability Design Documentation, https://developers.openai.com/api/docs/guides/tools-skills/

[30] OpenDSU, 2024. Anchoring (RFC-005), https://www.opendsu.org/pages/concepts/Anchoring-%28RFC-005%29.html

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

[31] OpenDSU, 2024. DSU Introduction (RFC-001), https://www.opendsu.org/pages/concepts/DSU-Introduction-%28RFC-001%29.html

[32] OpenDSU, 2026. Home / Overview, https://www.opendsu.org/

[33] OpenLineage, 2026. Facets & Extensibility / Dataset Facets, https://openlineage.io/docs/spec/facets/

[34] SPDX, 2026. Project overview and specification status, https://spdx.dev/

[35] SLSA, 2026. Provenance, https://slsa.dev/provenance

[36] Wang, L., et al., 2023. A Survey on Large Language Model Based Autonomous Agents, https://www.researchgate.net/publication/373297758_A_Survey_on_Large_Language_Model_based_Autonomous_Agents

[37] W3C, 2022. Decentralized Identifiers (DIDs) v1.0. W3C Recommendation, https://www.w3.org/TR/did/

[38] Wu, Q., et al., 2023. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation, https://www.researchgate.net/publication/373164024_AutoGen_Enabling_Next-Gen_LLM_Applications_via_Multi-Agent_Conversation_Framework.

[39] Yao, S., et al., 2023. ReAct: Synergizing Reasoning and Acting in Language Models, https://research.google/blog/react-synergizing-reasoning-and-acting-in-language-models/.

[40] European Commission, 2020, A European strategy for data, https://digital-strategy.ec.europa.eu/en/policies/strategy-data

[41] European Commission, 2025, European Data Union Strategy, https://digital-strategy.ec.europa.eu/en/policies/data-union

[42] Regulation (EU) 2023/2854 of the European Parliament and of the Council of 13 December 2023 on harmonised rules on fair access to and use of data and amending Regulation (EU) 2017/2394 and Directive (EU) 2020/1828 (Data Act), OJ L, 2023/2854, 22.12.2023, https://eur-lex.europa.eu/eli/reg/2023/2854/oj/eng

[43] European Commission, 2025, Common European Data Spaces, https://digital-strategy.ec.europa.eu/en/policies/data-spaces.

[44] European Commission (data.europa.eu), 2023, European data spaces and the role of data.europa.eu, https://data.europa.eu/sites/default/files/report/Data_Spaces_Panel_Report_EN.pdf

[45] International Data Spaces Association, 2025, How the EU Data Act will shape data spaces, https://internationaldataspaces.org/how-the-eu-data-act-will-shape-data-spaces/

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

[46] European Commission, 2022, Staff working document on data spaces, https://digital-strategy.ec.europa.eu/en/library/staff-working-document-data-spaces.

[47] Regulation (EU) 2025/327 of the European Parliament and of the Council of 11 February 2025 on the European Health Data Space and amending Directive 2011/24/EU and Regulation (EU) 2024/2847, OJ L, 2025/327, 5.3.2025, https://eur-lex.europa.eu/eli/reg/2025/327/oj/eng

[48] European Commission, 2025, European Health Data Space Regulation (EHDS), https://health.ec.europa.eu/ehealth-digital-health-and-care/european-health-data-space-regulation-ehds_en.

[49] Organisation for Economic Co-operation and Development (OECD), 2025, AI Risk Ontology (AIRO), OECD AI Catalogue of Tools & Metrics for Trustworthy AI, https://oecd.ai/en/catalogue/tools/airo-ai-risk-ontology

[50] Mitchell, M., Wu, S. et al., 2019, *Model Cards for Model Reporting*, Proceedings of FAT* 2019, https://arxiv.org/abs/1810.03993

[51] European Commission, 2024, *EU AI Act – Annex IV: Technical Documentation Requirements*, AI Act Service Desk, https://ai-act-service-desk.ec.europa.eu/en/ai-act/annex-4

[52] World Wide Web Consortium (W3C), 2024, Data Privacy Vocabulary (DPV), https://www.w3.org/community/dpv/

[53] European Union, 2024, Regulation (EU) 2024/1689 (Artificial Intelligence Act), https://eur-lex.europa.eu/eli/reg/2024/1689/oj

[54] World Wide Web Consortium (W3C), 2013, PROV-DM: The PROV Data Model, https://www.w3.org/TR/prov-dm/

[55] World Wide Web Consortium (W3C), 2020, Data Catalog Vocabulary (DCAT) Version 2, https://www.w3.org/TR/vocab-dcat-2/

[56] Neumaier, S., Dam, T., Kovac, F., 2024. *Semantic Web Journal manuscript (swj4042): Semantic interoperability and AI documentation*. Semantic Web Journal (IOS Press), https://www.semantic-web-journal.net/system/files/swj4042.pdf

[57] European Center for Not-for-Profit Law (ECNL), 2023. *Guide on Fundamental Rights Impact Assessments (FRIA),* https://ecnl.org/publications/guide-fundamental-rights-impact-assessments-fria

[58] European Union, 2016. *General Data Protection Regulation (GDPR), Regulation (EU) 2016/679, Article 35 – Data Protection Impact Assessment,* https://eur-lex.europa.eu/eli/reg/2016/679/oj

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.

[59] Amazon Web Services, 2025. *Firecracker MicroVM Documentation,* https://firecracker-microvm.github.io/

[60] Cloud Native Computing Foundation, 2025. *containerd Documentation,* https://containerd.io/

[61] Linux Containers Project, 2025. *LXC Documentation,* https://linuxcontainers.org/lxc/

Funded by the European Union

This project has received funding from the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101189689.