DRAFT International Standard

ISO/IEC DIS 26450

Information technology — Eclipse Dataspace Protocol (DSP)

ICS: 35.030

ISO/IEC JTC **1**

Secretariat: **ANSI**

Voting begins on:
2026-02-04

Voting terminates on:
2026-04-29

THIS DOCUMENT IS A DRAFT CIRCULATED FOR COMMENTS AND APPROVAL. IT IS THEREFORE SUBJECT TO CHANGE AND MAY NOT BE REFERRED TO AS AN INTERNATIONAL STANDARD UNTIL PUBLISHED AS SUCH.

IN ADDITION TO THEIR EVALUATION AS BEING ACCEPTABLE FOR INDUSTRIAL, TECHNOLOGICAL, COMMERCIAL AND USER PURPOSES, DRAFT INTERNATIONAL STANDARDS MAY ON OCCASION HAVE TO BE CONSIDERED IN THE LIGHT OF THEIR POTENTIAL TO BECOME STANDARDS TO WHICH REFERENCE MAY BE MADE IN NATIONAL REGULATIONS.

RECIPIENTS OF THIS DRAFT ARE INVITED TO SUBMIT, WITH THEIR COMMENTS, NOTIFICATION OF ANY RELEVANT PATENT RIGHTS OF WHICH THEY ARE AWARE AND TO PROVIDE SUPPORTING DOCUMENTATION.

This document is circulated as received from the committee secretariat.

Reference number
ISO/IEC DIS 26450:2026(en)

© ISO/IEC 2026



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2026

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Dataspace Protocol 2025-1

12 November 2025

▼ **More details about this document**

Latest published version:
https://eclipse-dataspace-protocol-base.github.io/DataspaceProtocol/2025-1-err1

Latest editor's draft:
https://eclipse-dataspace-protocol-base.github.io/DataspaceProtocol/

History:
Commit history

Editor:
Sebastian Steinbuss (International Data Spaces Association)

Authors:
Peter Koen (Microsoft)
Maarten Kollenstart (TNO)
James Marino (Metaform Systems)
Julia Pampus (Fraunhofer ISST)
Anil Turkmayali (International Data Spaces Association)
Sebastian Steinbuss (International Data Spaces Association)
Arno Weiß (SAP)

Feedback:
GitHub eclipse-dataspace-protocol-base/DataspaceProtocol (pull requests, new issue, open issues)

This document is licensed under The Apache License, Version 2.0.
Copyright © 2025 The Eclipse Foundation AISBL

Abstract

The Dataspace Protocol is a specification designed to facilitate interoperable data sharing between entities governed by usage control and based on Web technologies. This specification defines the schemas and protocols required for entities to publish data, negotiate Agreements, and access data as part of a federation of technical systems termed a Dataspace.

Status of this document

This version (2025-1) of the Dataspace Protocol specification is a release of the specification and considered to be stable. Further changes shall not affect conformity. All changes made to the specification can be reviewed in the GitHub repositories - up to and including `2024-1` under the governance of the International Data Spaces Association and with the Eclipse Dataspace Working Group ever since.

Table of Contents

Abstract

Status of this document

1. Introduction

2. Scope

3. Terms and definitions

4. Common Requirements
4.1 Authorization
4.2 Schemas & Contexts
4.3 Exposure of Versions
4.3.1 HTTPS Binding
4.4 Discovery of Service Endpoints

© ISO/IEC 2026 – All rights reserved

5. Catalog Protocol
5.1 Introduction
5.2 Message Types
5.2.1 Catalog Request Message
5.2.2 Dataset Request Message
5.3 Response Types
5.3.1 ACK - Catalog
5.3.2 ACK - Dataset
5.3.3 ERROR - Catalog Error
5.4 Technical Considerations
5.4.1 Queries and Filter Expressions
5.4.2 Replication Protocol
5.4.3 Security
5.4.4 Catalog Brokers

6. Catalog HTTPS Binding
6.1 Introduction
6.1.1 Prerequisites
6.1.2 Catalog Error
6.2 Path Bindings
6.2.1 Catalog Request Endpoint
6.2.2 Dataset Request Endpoint
6.3 Technical Considerations
6.3.1 Pagination
6.3.2 Compression

7. Contract Negotiation Protocol
7.1 Introduction
7.1.1 States
7.1.2 State Machine
7.2 Message Types
7.2.1 Contract Request Message
7.2.2 Contract Offer Message
7.2.3 Contract Agreement Message
7.2.4 Contract Agreement Verification Message
7.2.5 Contract Negotiation Event Message
7.2.6 Contract Negotiation Termination Message
7.3 Response Types
7.3.1 ACK - Contract Negotiation
7.3.2 ERROR - Contract Negotiation Error

8. Contract Negotiation HTTPS Binding
8.1 Introduction
8.1.1 Prerequisites
8.1.2 Contract Negotiation Error
8.1.3 Authorization
8.2 Provider Path Bindings
8.2.1 Contract Negotiation Endpoint
8.2.2 Contract Request Endpoint (init)
8.2.3 Contract Request Endpoint
8.2.4 Contract Negotiation Event Endpoint
8.2.5 Contract Agreement Verification Endpoint
8.2.6 Contract Negotiation Termination Endpoint
8.3 Consumer Path Bindings
8.3.1 Prerequisites
8.3.2 Contract Negotiation Endpoint
8.3.3 Contract Offer Endpoint (init)
8.3.4 Contract Offer Endpoint
8.3.5 Contract Agreement Endpoint
8.3.6 Contract Negotiation Event Endpoint
8.3.7 Contract Negotiation Termination Endpoint

© ISO/IEC 2026 – All rights reserved

9. Transfer Process Protocol
9.1 Introduction
9.1.1 Prerequisites
9.1.2 States
9.1.3 State Machine
9.2 Message Types
9.2.1 Transfer Request Message
9.2.2 Transfer Start Message
9.2.3 Transfer Suspension Message
9.2.4 Transfer Completion Message
9.2.5 Transfer Termination Message
9.3 Response Types
9.3.1 ACK - Transfer Process
9.3.2 ERROR - Transfer Error

10. Transfer Process HTTPS Binding
10.1 Introduction
10.1.1 Prerequisites
10.1.2 Transfer Error
10.2 Provider Path Bindings
10.2.1 Transfer Process Endpoint
10.2.2 Transfer Request Endpoint
10.2.3 Transfer Start Endpoint
10.2.4 Transfer Completion Endpoint
10.2.5 Transfer Termination Endpoint
10.2.6 Transfer Suspension Endpoint
10.3 Consumer Callback Path Bindings
10.3.1 Prerequisites
10.3.2 Transfer Start Endpoint
10.3.3 Transfer Completion Endpoint
10.3.4 Transfer Termination Endpoint
10.3.5 Transfer Suspension Endpoint

11. Conformance

A. Lower Level Type Definitions

B. Notes
B.1 List of Figures

C. References
C.1 Normative references
C.2 Informative references

1. Introduction

The **Dataspace Protocol** is used in the context of Dataspaces as described and defined in the subsequent sections with the purpose to support *interoperability*. In this context, the specification provides fundamental technical interoperability for Participants in Dataspaces.

This specification builds on protocols located in the ISO OSI model (ISO/IEC 7498-1:1994) layers, like the Hypertext Transfer Protocol (HTTP). The purpose of this specification is to define interactions between systems independent of such protocols, but describing how to implement it in an unambiguous and extensible way. To do so, the messages that are exchanged during the process are described in this specification and the states and their transitions are specified as state machines, based on the key terms and concepts of a Dataspace. On this foundation the bindings to Data Transfer Protocols, like Hypertext Transfer Protocol Secure (HTTPS), are described.

Note: This specification does not cover the data transfer as such. While this is controlled by the Transfer Process Protocol, e.g., the initiation of the transfer channels or their decomissioning, the data transfer itself and especially the handling of technical exceptions is an obligation to the transport protocol.

© ISO/IEC 2026 – All rights reserved

The classes and definitions used in this specification are reused from different standards and specifications as much as possible, in particular, Data Catalog Vocabulary (DCAT) [*vocab-dcat-3*] and Open Digital Rights Language (ODRL) [*odrl-model*]. As, however, the external definitions allow different interpretations or provide more attributes than required, this specification is leveraging Profiles of the original definitions rather than the complete original expressiveness. However, not every standard-compliant class might be compliant to the Dataspace Profile. They are not separate artifacts but implicitly contained in the JavaScript Object Notation (JSON) schemas for the Message Types of this specification.

This specification is organized into the following documents:

- 3. Terms and definitions documents define key terms.
- 4. Common Requirements declares cross-cutting functions as, e.g., the declaration of supported versions of this protocol and common data processing rules.
- 5. Catalog Protocol and 6. Catalog HTTPS Binding define how Catalogs are published and accessed as HTTPS endpoints respectively.
- 7. Contract Negotiation Protocol and 8. Contract Negotiation HTTPS Binding documents that define how Contract Negotiations are conducted and requested via HTTPS endpoints.
- 9. Transfer Process Protocol and 10. Transfer Process HTTPS Binding documents that define how Transfer Processes using a given Data Transfer Protocol are governed via HTTPS endpoints.

§ 2. Scope

Sharing data between autonomous entities requires the provision of metadata to facilitate the transfer of Datasets by making use of a data transfer (or application layer) protocol.

This document gives guidance of how this metadata is provisioned:

- This document specifies how Datasets are advertised via Catalogs (reusing terminology from [*vocab-dcat-3*]).
- This document specifies how data usage requirements are expressed as Policies (reusing terminology from [*odrl-model*] vocabulary).
- This document specifies how Agreements that govern data usage are syntactically expressed and electronically negotiated during a Contract Negotiation.
- This document specifies how Datasets are accessed using Transfer Process Protocols.

This document does not apply to the Data Transfer Protocol.

§ 3. Terms and definitions

The following terms and definitions introduce the core concepts, entities, and relationships that underpin a Dataspace and its communication protocols. Not all Dataspace entities have a concrete *technical* materialization; some entities *MAY* exist as purely logical constructs.

Agreement

A concrete Policy associated with a specific Dataset that has been agreed between the Provider and Consumer. It is a result of a Contract Negotiation defining the Policy agreed to for a Dataset.

Catalog

A collection of entries representing Offers that are advertised by a Provider.

Catalog Protocol

A set of allowable Message Types that are used to request a Catalog from a Catalog Service.

Catalog Service

A Participant Agent that makes a Catalog available and accessible to Participants.

Connector (Data Service)

© ISO/IEC 2026 – All rights reserved

A Participant Agent that performs Contract Negotiation and Transfer Process operations with other Connectors, by implementing Dataspace Protocols. It produces Agreements and manages Dataset sharing.

Consumer

A Participant that requests access to an offered Dataset.

Contract Negotiation

A set of interactions between a Provider and Consumer that establish an Agreement. It is an instantiation of the state machine of a Contract Negotiation Protocol. An outcome of a Contract Negotiation *MAY* be the production of an Agreement.

Contract Negotiation Protocol

A set of allowable Message Type sequences defined as a state machine.

Dataset

Data or a technical service that can be shared by a Participant.

Dataspace

governance framework and supporting services to build trustworthiness and enable the sharing of data through an agreed set of Policies, semantic models, protocols and processes

[SOURCE: ISO/IEC 20151 to be published]

Note 1 to entry: This specification convers only protocols to facilitate interoperable Dataset sharing between entities based on technical services. The overarching concepts and characteristics of Dataspaces are described in ISO/IEC 20151.

Dataspace Protocol

A set of Messages and Message sequences that enables the interaction between Participant Agents in a Dataspace. This may require additional concepts, which are not in the scope of the specification itself.

Data Transfer Protocol

A set of rules and conventions that dictate how data is transmitted over a network by defining the format, error handling, and flow control. Examples include HTTP, FTP, MQTT, and AMQP.

Message Type

A definition of the structure of a Message.

Offer

A concrete Policy associated with a specific Dataset.

Participant

A member of one or more Dataspaces that provides and/or consumes Datasets. It registers Participant Agents that perform tasks on its behalf.

Participant Agent

A technology system that performs operations and interactions in a Dataspace on behalf of a Participant, such as publishing a Catalog or engaging in a Transfer Process. It is a logical construct and does not necessarily correspond to a single runtime process. While most interactions take place between so-called Connectors, some interactions with other systems are required.

Policy

A set of rules, duties, and obligations that define the terms of use for a Dataset.

Profile

A restriction or subset of a specification that enforces every occurrence of an externally defined class to be conformant with the original definition.

Provider

A Participant that offers a Dataset.

© ISO/IEC 2026 – All rights reserved

Transfer Process

A set of interactions between a Provider and Consumer that give access to a Dataset under the terms of an Agreement. It is an instantiation of the state machine of a Transfer Process Protocol.

Transfer Process Protocol

A set of allowable Message Type sequences defined as a state machine.

4. Common Requirements

4.1 Authorization

All requests to HTTPS endpoints *SHOULD* use the `Authorization` header to include an authorization token. The semantics of such tokens are not part of this specification. The `Authorization` HTTP header is *OPTIONAL* if the Connector does not require authorization.

4.2 Schemas & Contexts

All protocol messages are normatively defined by a [*json-schema*]. This specification also uses JSON-LD 1.1 and provides a JSON-LD context to serialize data structures and Message types as it facilitates extensibility. The JSON-LD context is designed to produce Message serializations using compaction that validate against the JSON Schema for the given Message type. This allows implementations to choose whether to process messages as plain JSON or as JSON-LD and maintain interoperability between those approaches. Profiles that use JSON-LD are encouraged to provide similar contexts that facilitate this approach to interoperability. As this specification's JSON-LD objects are `@protected`, Profile authors are advised to define their custom terms as protected to spot term redefinition early.

4.3 Exposure of Versions

Connectors implementing the Dataspace Protocol *MAY* operate on different versions and bindings. Therefore, it is necessary that they can discover such information reliably and unambiguously. Each Connector *MUST* provide a version metadata endpoint ending with Uniform Resource Identifier (URI) segments `/.well-known/dspace-version`. The location of this endpoint *SHOULD* adhere to [*rfc8615*].

A Connector *MUST* respond to a respective HTTPS request by returning a `VersionResponse` with at least one item.

VersionResponse			
`protocolVersions`	*REQUIRED*	array[Version]	

The item connects the version tag (`version` attribute) with a path to the endpoint. The semantics of the `path` property are specified by each protocol binding. The `serviceId` is a unique identifier for a Data Service and allows to group Dataspace Protocol endpoints exposed by different Data Services across versions. `binding` describes the Dataspace Protocol binding such as HTTPS. `identifierType` describes the type of identifier used to refer to participants in the protocol communication.

Version			
`binding`	*REQUIRED*	string	Must be of the following: `HTTPS`
`path`	*REQUIRED*	string	
`version`	*REQUIRED*	any	
`auth`	*OPTIONAL*	Auth	

© ISO/IEC 2026 – All rights reserved