

Data Type Registry Interoperability Guideline

	Name	Partner/Activity	Date
From:	Hans Lienhop	GWDG	25.05.2025

Data Type Registry Interoperability Guideline.....	1
1. Introduction.....	2
1.1 Licensing Information.....	2
1.2 Intended Audience.....	2
1.3 Description and Main Features.....	2
2 Response to Community Need.....	3
3 High-level Architecture.....	3
3.1 Definitions.....	4
4 Related Guidelines.....	4
5 Adopted Standards.....	4
6 Integration Options.....	5
6.1 Integration Option 1: Integration via PID.....	5
6.2 Integration Option 2: Search Integration.....	5
6.3 Integration Option 3: Integration of Data Validation.....	5
6.4 Cross DTR Interoperability.....	6
6.4.1 Technical Interoperability.....	6
6.4.2 Semantic Interoperability.....	6
6.4.3 Organisational Interoperability.....	6
6.4.4 Legal Interoperability.....	6
7 Conclusion.....	6



1. Introduction

1.1 Licensing Information

These guidelines are developed under the FAIRCORE4EOSC project and align with the EOSC Interoperability Framework (EOSC-IF). Refer to ¹ and ² for licensing information regarding the software components.

1.2 Intended Audience

These guidelines target technical experts, software developers, research infrastructure managers, and data service providers who intend to implement or integrate a Data Type Registry (DTR) compliant with EOSC Interoperability standards.

1.3 Description and Main Features

The Data Type Registry (DTR) is a key service developed as part of the FAIRCORE4EOSC project. It provides a framework for the registration, discovery, and management of structured data type definitions. The DTR enables communities to define and publish reusable, machine-readable data types, thereby supporting better data interoperability and reuse in line with the FAIR principles.

The registry is built on top of CNRI's Cordra platform³, which offers a robust infrastructure for managing digital objects. Each data type is represented as a digital object, described in JSON, and assigned a persistent identifier (PID). This approach ensures that type definitions are not only structured, but also globally addressable and resolvable over time.

The core goal of the DTR is to offer a sustainable, open service that supports the long-term curation and referencing of data type definitions. It is designed to help data providers ensure that their data structures are well-documented and can be easily understood and reused by others. At the same time, it facilitates discovery and validation by researchers, developers, and service providers who require standardized ways of interpreting data.

The registry is complemented by additional software components, such as the TypeAPI, and is supported by clear governance guidelines. These elements together form a complete ecosystem that addresses the technical, organizational, and semantic challenges of managing data types across distributed research infrastructures.

This document describes guidelines on how to incorporate the DTR into other projects, as well as addressing four interoperability layers to enable cross-DTR interoperability:

- **Technical Interoperability:** Standardization of APIs, protocols, identifiers, and data formats.
- **Semantic Interoperability:** Use of common vocabularies, ontologies, and metadata standards.
- **Organisational Interoperability:** Governance and community engagement aligned with EOSC principles.
- **Legal Interoperability:** Compliance with GDPR and data licensing requirements.

¹ <https://gitlab.com/cnri/cordra/cordra/blob/master/LICENSE.txt>

² <https://github.com/FC4E-T4-3/type-api/blob/main/LICENSE>

³ <https://www.cordra.org/>

2 Response to Community Need

The new version of the Data Type Registry addresses the need for a standardized approach to manage and share data type definitions across research domains.

3 High-level Architecture

The architecture of the DTR is designed to be modular and extensible, allowing various components to interact seamlessly while maintaining clear responsibilities. The system leverages established technologies to ensure persistence, scalability, and flexibility in managing structured data types.

Each data type in the registry is treated as a digital object, managed within the Cordra platform. These objects are assigned persistent identifiers, versioned, and stored with structured metadata. The system supports REST and DOIP APIs, providing both human-friendly and programmatic access to type definitions. On top of this core, the TypeAPI as additional service enhances the registry's capabilities for schema generation, federated discovery, and validation.

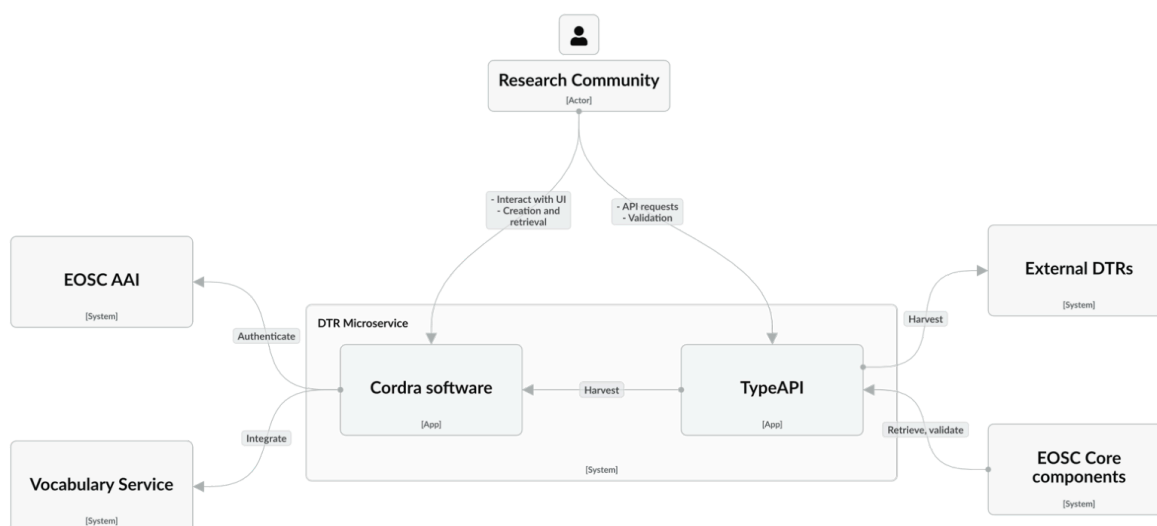


Figure 1 - High Level Architecture

Cordra: The Cordra platform serves as the backbone of the DTR infrastructure. It is a highly adaptable system for managing JSON-based digital objects, each of which is identified by a persistent identifier registered through the Handle system. Cordra was selected for its robust support for PID management and customizable object modelling.

TypeAPI: To support advanced functionality such as schema generation, validation, and federated discovery, the DTR includes a dedicated service called the TypeAPI. This component is implemented as a RESTful web service.

The TypeAPI extends the capabilities of the DTR beyond simple object storage by providing programmatic access to type definitions in a format suitable for schema processing and automated workflows. One of its core functions is the transformation of DTR type descriptions into JSON Schema, which allows developers and services to validate data instances against a formally defined structure.

3.1 Definitions

Term	Definition
Data Type	A data type is a reusable, uniquely identified definition that describes the structure, semantics, and constraints of a specific kind of data element—typically used within metadata schemas to ensure consistency, interoperability, and machine-actionability across systems.
Type Description	The Type Description of a data type is a JSON object registered in the DTR, that describes the properties of the data type
Schema	A JSON Schema conforming representation of a data type describing schema elements.

Table 1 - Definitions

4 Related Guidelines

ResourceType	Title	Short Description	Reference
Guideline	EOSC Interoperability Framework (EOSC-IF)	An EOSC provided framework for the general structure of Interoperability guidelines.	https://op.europa.eu/s/z5m1
Guideline	EOSC Persistent Identifier (PID) Policy	An EOSC policy for the use and provision of persistent identifiers.	https://op.europa.eu/s/z5m0
Guideline	EOSC Authentication and Authorization Infrastructure (AAI)	Description of the EOSC AAI architecture, principles, and guidelines	https://op.europa.eu/s/z5mZ

Table 2 - Related Guidelines

5 Adopted Standards

Standard	Description	Reference
REST	All the APIs exposed by the DTR follow the REST standard	https://www.w3.org/2001/sw/wiki/REST
JSON-LD	Metadata format for semantic interoperability	https://www.w3.org/TR/json-ld11/
OAuth 2.0	Used to authenticate/authorise users in the Web interfaces.	https://oauth.net/2/
JSON Schema	Used for validation of data for compliance with data types.	https://json-schema.org/
HTTPS	All Web Services of the DTR are exposed via the HTTPS	https://tools.ietf.org/html/rfc2818

Table 3 - Adopted Standards



6 Integration Options

In this section, we will differentiate between two different kinds of integration. On one hand, we will describe how data types registered in the DTR can be integrated into other services. On the other hand, we will provide information on what is needed for a self-hosted DTR to be considered interoperable with other DTRs.

6.1 Integration Option 1: Integration via PID

The persistent identifiers (PIDs) of the data types are designed to be resolvable, for example, through the global handle resolver⁴. They serve as stable references and can be used independently to identify data types across various scenarios. By default, a PID resolves to a user interface view of the type description; however, by appending the parameter *locatt=view:json* to the URL, it is possible to directly access the JSON representation of the type. For instance, the request <https://hdl.handle.net/21.111969/11f856bc964aa76624f2?locatt=view:json> would lead to the JSON view.

6.2 Integration Option 2: Search Integration

Services have the option to integrate direct search functionalities for discovering and utilizing data types. Individual Data Type Registries (DTRs) can be queried using the search capabilities provided by the Cordra software. For a broader scope, federated search can be implemented using the TypeAPI's search endpoint, which allows simultaneous querying across all integrated registries. The search functionality supports several parameters, including *query*, *queryBy[]*, *filterBy*, *pageNum*, *pageSize*, and *infix*. For example, the following query searches for the term "test" within the name, authors, and description fields for all registered types in the federation while enabling infix matching:

<https://typeapi.pidconsortium.net/v1/search?query=test&queryBy=name&queryBy=authors&queryBy=description&date=1744070400&infix=true>.

6.3 Integration Option 3: Integration of Data Validation

The TypeAPI component of the Data Type Registry (DTR) supports the generation of JSON schemas that describe specific data types. It also provides an endpoint for direct data validation, allowing services to verify whether given data conforms to a specified type. This is achieved by sending a POST request with the data to be validated to the appropriate TypeAPI instance. For example, data can be validated against a specific type by sending a request to:

<https://typeapi.pidconsortium.net/v1/types/validate/21.111969/d4242fb8297d3ff4199b>,

where the payload is the JSON data subject to validation.

⁴ <https://hdl.handle.net/>

6.4 Cross DTR Interoperability

6.4.1 Technical Interoperability

To ensure technical interoperability across Data Type Registries (DTRs), all software implementations must expose RESTful APIs with clearly defined endpoints for registering, retrieving, and updating content. These implementations must comply with the EOSC Persistent Identifier (PID) Policy, using persistent identifiers, for example Handle identifiers, to uniquely reference all relevant entities requiring shared access, including data type schemas. For metadata exchange, the use of JSON-LD and JSON Schema is mandated. Secure authentication and authorization mechanisms should be implemented using OAuth2.0 or SAML2.0, and user access control must align with the EOSC Authentication and Authorization Infrastructure (AAI).

6.4.2 Semantic Interoperability

Semantic interoperability is achieved by adhering to the FAIR principles, ensuring that all metadata is machine-readable and that any recreations of established data types are properly linked to corresponding ontologies. Metadata schema harmonization must also be enforced, requiring the use of standardized minimal schemas⁵ to describe data types consistently across registries.

6.4.3 Organisational Interoperability

Organisational interoperability is guided by the governance structures defined for the DTR. These structures outline the responsibilities for maintaining standards and handling updates within the DTR ecosystem and compliance is a requirement to achieve cross-DTR interoperability. Registered data types must conform to the EOSC Rules of Participation⁶, particularly regarding interoperability criteria. Furthermore, collaboration between registries is encouraged to promote compatibility and alignment across different DTR instances.

6.4.4 Legal Interoperability

Legal interoperability requires that all metadata stored within the DTR be published under permissive licenses, specifically CC0 or CC BY 4.0. Clear documentation of usage rights and obligations must be provided for each registered data type. To comply with the General Data Protection Regulation (GDPR), personal data must be anonymized, and explicit consent mechanisms should be in place. Additionally, legal frameworks recommended by EOSC should be followed to support secure and lawful cross-border data sharing.

7 Conclusion

By adhering to these interoperability guidelines, the Data Type Registry (DTR) developed within the FAIRCORE4EOSC project will promote seamless data type management and foster cross-disciplinary data sharing within EOSC, supporting an open, FAIR, and federated research ecosystem.

⁵ <https://github.com/FC4E-T4-3/schemas>

⁶ <https://op.europa.eu/s/z5m3>