International FAIR Digital Objects Implementation Summit 2024

Extended Abstracts

https://doi.org/10.52825/ocp.v5i.1197

Published: 18 Mar. 2025

Meet the FDO Manager

Reference Implementation of FDO Operations (Create, Retrieve, Update, Delete) Based on the Digital Object Interface Protocol (DOIP).

Henrik tom Wörden[1,*], Timm Fitschen[1], Sven Bingert[2], and Tibor Kálmán[2]

[1]IndiScale GmbH, Germany

[2]Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen, Germany

*Correspondence: Henrik tom Wörden, h.tomwoerden@indiscale.com

Abstract. The FDO Manager is a free and open source reference implementation of basic FDO Operations like retrieve, create or update. It is based on the Digital Object Architecture and realizes an FDO architecture employing persistent identifiers that can be resolved via the Digital Object Identifier Resolution Protocol.

Keywords: FAIR Digital Objects, Handle System, Data Space, DOIP

1. Introduction

The FDO Manager is a free and open source reference implementation of FDO Operations [1] (create, retrieve, update, delete) based on the Digital Object Architecture (DOA) and using the Digital Object Interface Protocol (DOIP) [2] according to the latest FDO specifications [3]. This implementation realizes an FDO architecture by augmenting the DOA to identify possible new extensions of the DOIP or existing software for the DOA.

2. Implementation

In full compliance with the DOA, an FDO is identified by a single Persitent Identifier (PID) Record resolvable via the Digital Object Identifier Resolution Protocol (DO-IRP) [4]. The PID Record of the FDO references two plain Digital Objects, one is the FDO's data and one is the FDO's meta data. Furthermore, it holds the FDO Profile and the FDO Type. FDO-enabled clients and services, such as the FDO Manager itself, can validate the FDO-ness of a plain Digital Object (DO) through a duck typing mechanism.

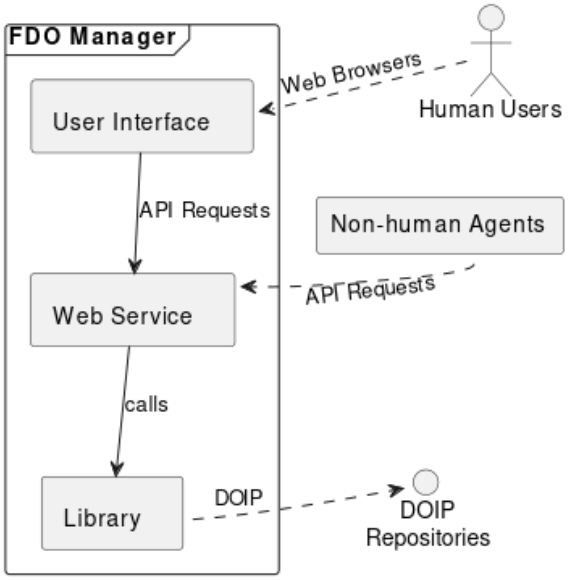

Figure 1. *Architecture of the FDO Manager*

The FDO Manager contains three components: (1) A software library written in Java that implements the operation logic. (2) A web service in Java offering a RESTful API documented as an OpenAPI Specification (OAS). (3) An intuitive web-based user interface (UI). The UI calls the RESTful API and the service uses the transaction library internally. Other clients and third-party software can use the API or implement own services based on the library. The FDO Manager supports creating FDOs in repositories, such as Cordra, from a registry of trusted DOIP repositories by uploading data files and meta data, retrieving and viewing FDO meta data, downloading FDO data files, moving or duplicating FDO data and meta data from one repository to another, updating FDO data and meta data, as well as deleting FDOs from particular repositories.

Our approach allows to reuse existing infrastructure, such as DOIP clients, repositories and the DO-IRP service. Every FDO is a DO, referencing other DOs through PIDs. On the other hand, not every DO is an FDO automatically. However, this approach allows to create FDOs from existing DOs at any time by adding meta data and an FDO Record or by updating the DO's PID Record to include the FDO Profile, FDO Type, as well as references to data and meta data.

This presentation introduces the FDO Manager to the FDO community. It highlights the current state of the implementation, where the code and documentation can be downloaded, how it can be tested, how interested parties can contribute. Furthermore, it discusses some lessons learned, the limitations of our approach and the plans for the near future.

Underlying and related material

https://gitlab.com/fairdo/fdo-manager/

Author contributions

T. Fitschen: Conceptualization, Methodology, Software, Visualization, Writing – original draft, Writing – review & editing

H. tom Wörden: Conceptualization, Methodology, Software, Writing – review & editing

S. Bingert: Conceptualization, Methodology, Software, Writing – review & editing

T. Kálmán: Conceptualization, Methodology, Software, Writing – review & editing

Competing interests

All authors declare that they have no competing interests.

Funding

This work is funded by the Federal Ministry for Digital and Transport as part of the MISSION KI - National Initiative for Artificial Intelligence and Data Economy.

References

[1] N. Blumenröhr et al., “Taxonomy of Operations in the Context of FAIR Digital Objects”, 2023, doi: 10.5281/zenodo.8124177.
[2] DONA Foundation, “Digital Object Interface Protocol Specification, version 2.0“, 2018, url: https://www.dona.net/sites/default/files/2018-11/DOIPv2Spec_1.pdf.
[3] I. Anders et al. “FAIR Digital Object Technical Overview. PEN 2.0”, 2023, doi: 10.5281/zenodo.7824714.
[4] DONA Foundation, “Digital Object Identifier Resolution Protocol (DO-IRP) Specification, version 3.0”, 2022, url: https://www.dona.net/sites/default/files/2022-07/DO-IRPV3.0--2022-06-30_0.pdf