FAIR **DIGITAL OBJECTS** FORUM

FAIR Digital Object
Technical Overview

Version PEN 2.0

FDO Forum Proposed Endorsed Note document

18. January 2023

Current and previous versions:

Current version:

Previous version:
https://docs.google.com/document/d/1GAj-1owAAPDF7hVis2dYPPCOHiPHrlXlo4j-S5AdEFI/edit?usp=sharing

Editors:

Larry Lannom, Ulrich Schwardmann, Peter Wittenburg, Christophe Blanchi, Ivonne Anders, Claus Weiland

Authors:

I. Anders, Ch. Blanchi, Daan Broeder, M. Hellström, Sh. Islam, Th. Jejkal, L. Lannom, K. Peters, R. Quick, A. Schlemmer, U. Schwardmann, Stian Soiland-Reyes, G. Strawn, D. van Uytvanck, C. Weiland, P. Wittenburg, Carlo Zwölf

Abstract

This document is a comprehensive overview of the work on FAIR Digital Objects. It addresses a variety of aspects that have been discussed in the FDO Forum during the last two years and that are the subject of formal FDO Forum documents. Many of these documents have been written including examples and illustrations and in a verbose style allowing many to follow the discussion. This document will present an overview and should be seen as complementary to the FDO Requirements Specification document which has been extended as a result of the specification work.

FAIR DIGITAL OBJECTS FORUM

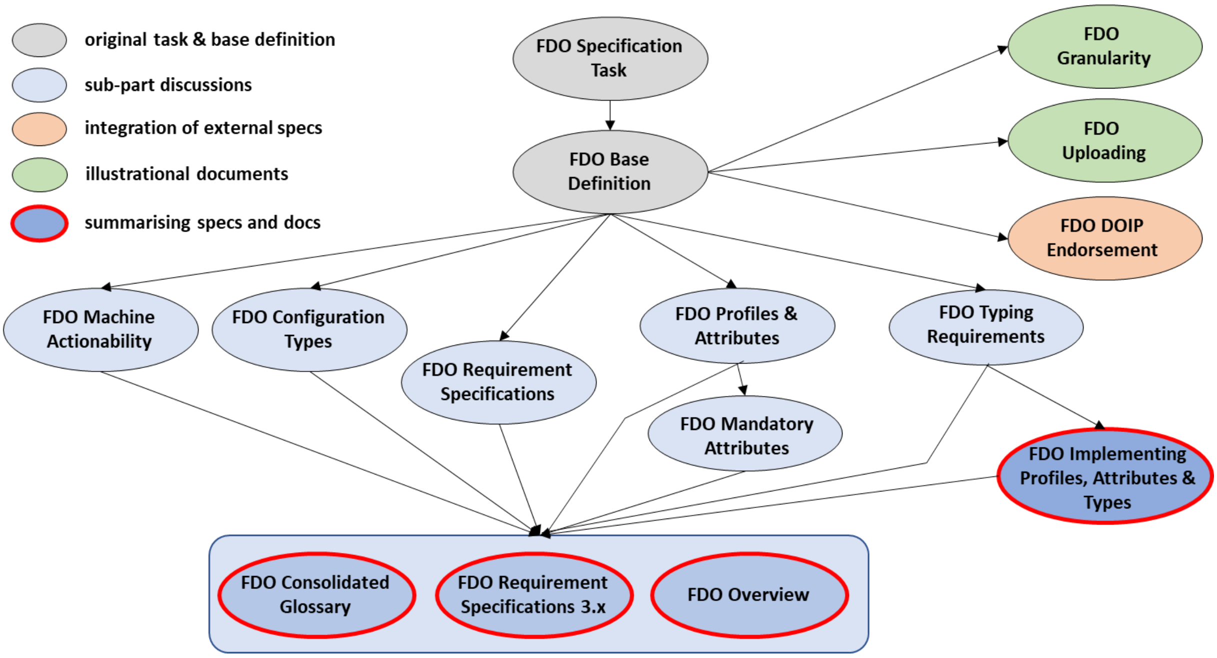
Status of this document

This document version is a Proposed Endorsement Note 2.0 since we split the previous document called "Full Specification Document" in an overview part (this document) and an extended FDO Requirements Specification PR3.0 document. This overview part does not contain binding specifications; therefore it is a Proposed Endorsement Note (PEN).

Guide to the FDO Documents

The specification task was started with a base definition of an FDO and then to work first on sub-aspects to make the specification task feasible. A set of specification documents evolved over time (light blue). In addition, two documents were discussed for illustrative purposes (green) and until now one external document was endorsed as important for FDO considerations (orange). The specifications so far are summarised in the documents in four documents (dark blue). It should be noted that the earlier written documents need a revision to make them compliant with the glossary as it is evolving.

FDO Machine Actionability explains what in the realm of FDO discussions is meant with machine actionability.
FDO Configuration Types explains what kind of variants FDOs can take which is important to understand for writing proper requirement specifications.
FDO Requirement Specifications were started at the Paris FDO meeting (2019) and were extended stepwise by new insights about FDOs.
FDO Profiles & Attributes and **FDO Mandatory Attributes** define requirements with respect to profiles and kernel attributes and state which attributes need to be mandatory in every profile.
FDO Typing Requirements make statements about the Typing System for FDOs.



FDO-Granularity is based on earlier EOSC documents and explains why granularity choices for FDOs are widely determined by the application domain.
FDO Uploading is an example of how FDOs and FDO collections can be integrated into the domain of digital objects.
FDO DOIP Endorsement suggests testing and adopting the DOIP V2.0 interface protocol as it has been worked out by the DONA Foundation.
FDO Glossary is an incrementally extended document which will contain explanations of all major terms.
FDO Requirement Specification 3.x is the document which contains all FDO requirements in a semi-formal style resulting from all documents that are being endorsed.
FDO Overview (this document) gives an overview about the FDO work.
FDO Implementing Profiles, Attributes & Types is the document that turns requirements into specifications that can be used for implementation purposes.

This document contains the following sections:
Section 1 contains easy to read descriptions to understand FDOs (FDO Model, Base Definition, Short History).
Section 2 summarises a set of core aspects extracted in short form from the various FDO specification documents (machine actionability, configuration variants, global PID system, profiles & attributes, typing system).
Section 3 includes relevant summaries of illustrating and endorsed documents.

All documents can be found here:
https://drive.google.com/drive/u/0/folders/1-SbZk7enOqjy2Rf57PMB-CQW7qMOUXgO

Formal References

This document is based on the following official FDO Documents[1] indicated by [FRx] in the text.

1. FDO Base Definition: see chapter 2
2. FDO Requirement Specifications:
https://docs.google.com/document/d/15M-OSlbaLrqyPUuVoo2sBv124_GwA6OfjS_xDIMMPTQ/edit?usp=sharing
3. FDO Configuration Types:
https://docs.google.com/document/d/1LojKBfuzuI6tKSMSZM77qd45lHmMFAxleZR7hl1DAUk/edit?usp=sharing
4. FDO Machine Actionability:
https://docs.google.com/document/d/1PPGW83siPDMLG5QRHsKM2cOUPT-O5SqS_jZvs_AO1Qw/edit?usp=sharing
5. FDO PID Profiles and Attributes:
https://docs.google.com/document/d/1iirLKoWiG9D2ZZ65RqVk89bCwSYwBVHBxk5gLmeNigl/edit?usp=sharing
6. Mandatory and and Optional Kernel Attributes:
https://docs.google.com/document/d/11CuSoNOpg3vYaoHqi4Yg6ncP3qOcUIC/edit?usp=sharing&ouid=105963437005859657844&rtpof=true&sd=true

[1] Final recommendation versions of all documents will be uploaded to a ZENODO folder and receive a DOI.

7. FDO Granularity, Versioning and Mutability:
https://docs.google.com/document/d/1jc7FRtzosJBHzkl5oVh-1USnYr3tB_RKDGrDq4uynUg/edit?usp=sharing
8. FDO Typing:
https://docs.google.com/document/d/1X0hcOVIqP7iYIJf9u-7x3RwcXK8ecsauy0FZg_6-Bg0/edit?usp=sharing
9. FDO Implementing Attributes, Profiles, Types (in progress):
https://docs.google.com/document/d/1bS1wtBjleZUWFZQuXlH2mD3H-gft7cj_bfY1USy5jgQ/edit?usp=sharing
10. FDO Upload Document:
https://docs.google.com/document/d/1NkkdilslxuCrvexSKtQXIPopMhE7nHOpd8X4m-3YJ5s/edit?usp=sharing
11. FDO DOIP Endorsement:
https://docs.google.com/document/d/1VJjpSztDm9HqR3zKrAeZXR55llorajkvfMUQ0HbbPTk/edit?usp=sharing

This document also refers to external specifications that have been endorsed by the FDO Forum or that are referenced by some of the above documents indicated by [Rx] in the text.

1. DO-IRP: https://www.dona.net/sites/default/files/2022-06/DO-IRPV3.0--2022-06-30.pdf
2. Handle System Specification: https://www.handle.net/
3. RDA Data Type Registry: https://www.rd-alliance.org/group/data-type-registries-wg/outcomes/data-type-registries
4. EOSC PID Document: https://op.europa.eu/en/publication-detail/-/publication/3136c3e6-4f07-11eb-b59f-01aa75ed71a1/language-en
5. GEDE PID Document: https://zenodo.org/record/1116189
6. DOI Specifications: https://www.doi.org/doi_handbook/pub_agreements/DOICoreSpecificationv1.pdf
7. RDA Data Core Model: http://hdl.handle.net/11304/5d760a3e-991d-11e5-9bb4-2b0aad496318
8. RDA Kernel Type Information: https://www.rd-alliance.org/groups/pid-kernel-information-profile-management-wg
9. RDA Research Collections: https://www.rd-alliance.org/groups/research-data-collections-wg.html
10. Research Objects and RO-Crate: https://www.researchobject.org/ro-crate/
11. Signposting Specifications: https://signposting.org/
12. Luiz Bonino's FDO Framework: https://fairdigitalobjectframework.org/
13. CoreTrustSeal: https://www.coretrustseal.org/

Since terminology varies considerably among these documents, we include in this work a glossary that defines a set of terms which are key for the understanding of FDOs and thus this document. Where feasible we relate the FDO terms to other terms being used.

Acknowledgments

This document is based on many discussions in all FDO Working Groups and therefore we thank all WG members for their contributions.

Contents

1. FAIR Digital Objects

1.1 Introduction

Today the data space is huge, heterogeneous and fragmented leading to 80% waste of time in data-driven projects in research and industry[2], reducing the effectiveness of research, hampering innovation, increasing costs and this issue will only grow over time. The massive and ubiquitous deployment of smart Internet of Things devices will lead to an even larger increase of the data space and its inherent complexity requiring the development of technical solutions to address it more efficiently to simply keep up. New disruptive technologies such as Quantum computing and DNA-based data storage may provide some solutions to these challenges while creating new ones with potentially dramatic consequences.

To address this challenge, there is a need to globally define the basis for convergence to address the increasing volume, complexity, and diversity of the dataspace. FAIR Digital Objects are simple, autonomous globally and uniquely identified, persistent, and consistently accessible units that propose a basic model for representing and interacting with data and services that could remain relevant across a wide range of technological innovation by acting as an island of stability for data in the coming decades if not longer.

To increase trust in the specifications and their further development there is a need to turn them into international standards and to clearly identify the organisation that takes responsibility and stands for accountability. The FDO Forum has begun this process by signing a collaboration agreement with DIN (German Institution for Standards) to turn all specifications to international standards in the realm of ISO.

1.2 Short History

The term “digital object” is well-known in computer science. It was first used in the realm of “Abstract Data Types” which then evolved to the principle of “object-oriented programming” which found its way into many state-of-the-art programming languages to support encapsulation and thus to make large software systems more robust. In 1995 Kahn & Wilensky wrote a first paper titled “A framework for distributed digital object services” where they used the term “digital object” in relation to data services in the Internet domain. This concept led to the Handle System enabling users to assign PIDs to digital objects which was then initially applied by the publishing industry to define the Digital

[2] For industrial projects a survey can be found in the Data Scientist Report, Crowdflower. (2017) https://visit.crowdflower.com/WC-2017-Data-Science-Report_LP.html

Object Identifiers (DOIs) which are Handles with a specific prefix and business model assigned at first to most electronic publications and since extended to movies, building materials, and other object types.

The concept of Digital Objects was then also used in database systems which have an object layer on top of the logic layer. Later the concept was applied by cloud systems using the term "object stores" to describe their layered system capable of handling vast amounts of data efficiently. Here each digital entity stored in the cloud is assigned a local identifier which gives access to administrative information (metadata) about the entity, including a reference to its physical location.

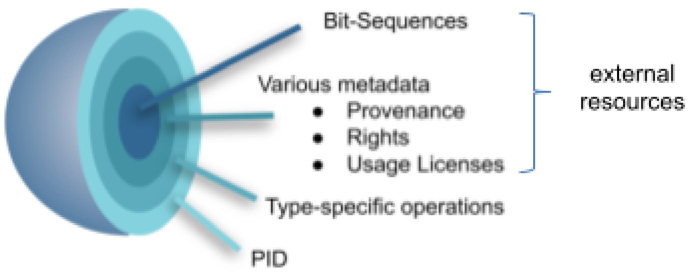
In 2014 the RDA Data Foundation & Terminology working group tested the Kahn & Wilensky model against a wide variety of practical cases and extended it to describe a core data model emphasising the principles of abstraction and binding [R7]. In 2014 the DONA Foundation was established and as part of its statutes it took responsibility to further the development and adoption of the Digital Object Architecture and its related standards such as the DO Interface Protocol (DOIP V2.0) [FR11] and DO Identifier Resolution Protocol (DO-IRPV3.0) [R1], an update of the older Handle System RFCs [R2].

In parallel the FAIR principles were emerging which resulted in the Nature paper in 2016 [??] which in the meantime led to their broad acceptance globally. In 2019 experts who developed the FAIR principles and those who were working on the core data model behind digital objects came together to discuss the requirements that could make digital objects FAIR compliant. This resulted in the initiative on FAIR Digital Objects which is being discussed in this document.

The Web community was also working to improve data management. Recently, the Signposting solution [R11] was suggested to implement the concept of "digital objects" by resolving URLs to specifically structured HTML pages that include references to all relevant information about a digital entity using IANA defined structures and semantics such as MIME types. Also, the concept of "Linked Data" has received a great deal of attention since it offers mechanisms for publishing and connecting structured data on the Web. Core element is a relation that is used to refer to other data on the web applying the Resource Description Framework (RDF).

1.3 FDO Model

The FAIR Digital Objects (FDO) provide a basic solution to the needs of a rapidly evolving global integrated data space in the same manner that the Datagram data model of TCP/IP solved the problem of interacting across various networks around the world.



The FDO Model

As the figure illustrates, an FDO consists of four layers[3]:

1. The PID System which resolves a global unique resolvable and persistent identifier (PID[4]) into a PID Record that may be a consistent FDO record[5] binding the necessary information to enable FDOs' Fairness.
2. The FDO record which defines a machine readable, interpretable, and actionable structure that specifies the FDO's type, its metadata, its machine interpretable FDO kernel attributes, and its sets of bit sequences either by instance of by reference according to the record's FDO profiles and the Kernel Attribute definitions.
3. The FDO resource layer provides access to external resources in a distributed landscape of trusted and FAIR compliant service providers. The FDO Record must contain valid references to these external resources.
4. A set of mandatory and recommended attributes. One of the mandatory attributes of the FDO record is the type of the FDO which provides a short hand for the characteristics of the FDO and relates the FDO to operations which are themselves FDOs.

These four layers are the foundation that make the FDO Model FAIR compliant. The FDO Profiles define the set and cardinality of kernel attributes used by a specific service provider who created the FDO record. To make the FDO Record machine actionable all FDO profiles and attributes be defined and registered in open registries.

A wide range of FDO compliant configurations, encapsulating data in a variety of structures such as databases, files, cloud systems, or other technologies that will emerge in the future. These may be described using different types of metadata (general, scientific, provenance, rights, usages, etc.) which are tightly and persistently bound by the information anchored in the FDO record.

Metadata is used here in its broad sense as containing statements about a variety of properties such as the typical descriptive metadata used for general searches, detailed scientific and technical metadata to support detailed research work and to facilitate automatic workflows including contextual and provenance descriptions, information about the rights to access and reuse the content, and information about transactions and contracts if this is required.

The persistent binding is achieved by assigning each FDO a globally unique resolvable and persistent identifier (PID) and associating the PID with a set of defined kernel attributes which are returned in a machine actionable form when a given PID is resolved.

In general, it is assumed that resources encapsulated by FDOs are managed and maintained in trustworthy repositories to guarantee long-term persistence and FAIRness. Suggestions to assess the quality of the procedures of repositories have been put forward, e.g., the CoreTrustSeal [R13], which resulted from collaborations in the Research Data Alliance.

FDOs will foster the stepwise increase of the trust level of the resource providers, which will be a vital and potentially lengthy process. FAIRness and adherence to rule systems such as CoreTrustSeal will

[3] General metadata and bit-sequences are managed by external providers, which is why we do not distinguish between these two although they have different functions as indicated by the FAIR principles, for example.
[4] We use the shortcut PID since it has been used in so many documents.
[5] PIDs can be used in other contexts where they resolve to PID records adhering to other regulations.

increase trust in FDOs. The encapsulation of access to the resources by using controlled operations is another important step. FDOs support the linking of data/information types with community defined operations so that common operations are easy and incorrect or disallowed operations are prevented.

1.3 Base Definition

The FDO Forum endorsed the following base definition and a slightly extended technical definition of a FAIR Digital Object.

A base definition
A FAIR Digital Object is a unit composed of data and/or metadata regulated by structures or schemas, and with an assigned globally unique and persistent identifier (PID), which is findable, accessible, interoperable and reusable both by humans and computers for the reliable interpretation and processing of the data represented by the object.

A full technical definition
A FAIR digital object is a unit composed of data that is a sequence of bits or a set of sequences of bits, each of the sequences being structured (typed) in a way that is interpretable by one or more computer systems, and having as essential elements an assigned globally unique and persistent identifier (PID), a type definition for the object as a whole and typically a metadata description (which itself can be another FAIR digital object) of the properties of the object, making the whole findable, accessible, interoperable and reusable both by humans and computers for the reliable interpretation and of the data represented by the object.

The FDO specification needs to be technologically neutral to enable different implementation approaches. This will facilitate interoperability across diverse implementations and will foster interests in adopting the technology when faced with the need for repository managers to make their data accessible to a wide range of repositories and services around the world.

Currently, two known FDO implementation approaches include the following:
1) Web based specifications that use URI/URLs for PIDs, and HTML / JSON and/or Linked Data as data model [6]
2) Digital Object Architecture based solution using Handles/DOIs as PIDs and Digital Objects as core data model.

2. Core Aspects

To understand an FDO it is important to first describe its core conceptual aspects described in the different official documents on FDO. The following sections summarise these core aspects.

2.1 Machine Actionability

As described in “MachineActionDef” [FR4] digital objects which are FAIR compliant and can be processed by machines without human intervention are called FAIR Digital Objects.

[6] See the Signposting specifications [R11]

1. There are as yet no clear conceptual distinctions and widely agreed definitions about the terms "machine readability", "machine interpretability" and "machine actionability".
2. For the purposes of this specification we agree on the following definitions:
    a. "machine readable" are those elements in bit-sequences that are clearly defined by structural specifications.
    b. "machine interpretable" are those elements that are machine readable and can be related with semantic artefacts in a given context and therefore have a defined purpose.
    c. "machine actionable" are those elements in bit-sequences that are machine interpretable and belong to a type for which operations have been specified in symbolic grammar.
3. We need to acknowledge that for FDOs there are organisational entities with PID records that have Kernel Attributes. After PID resolution, these attributes provide metadata information -- either directly or by pointing to metadata descriptions that need interpretation. Therefore, the FDO machine actionability in the sense of FAIRness requires three levels of checks:
    a. Does the resolution of a PID happen according to a standardised protocol and is the resolution result predictable and appropriate according to a specified profile?
    b. Does the resulting resolution entail machine actionable attributes, i.e., can the attributes be interpreted and lead to actions based on clear definitions and typing? Are the results obtained machine actionable, i.e., can the attributes be interpreted and lead to actions based on clear definitions and typing? This must include references to essential metadata descriptions (descriptive, scientific, rights/licences, access permissions, etc.).
    c. Are the metadata descriptions accessible and FAIR compliant, i.e., that in the full sense of FAIR all specifications[7] are machine actionable if they are embedded in an FDO infrastructure that implements policies, rules and procedures for FAIR?

It should be noted that 3c will be difficult to turn into practice since metadata will be created and managed by different communities and it will not be simple to change community practices. Therefore, the FDO Forum needs to find mechanisms to bridge between requirements and practices.

2.2 Global PID System

Any PID[8] system to be used to identify FDO should be global, robust, scalable, persistent, secure, and resistant to attacks is at the core of the FDO's trust pyramid. A FDO PID is associated with information that is structured in a PID record. A PID record is accessed by resolving an FDO PID. The information associated with the PIDs needs (a) to be specified by an FDO Profile, (b) to contain defined and registered metadata called Kernel Attributes, (c) to be secured using a public key infrastructure, and if needed standard certificates, to insure that only the PID owner can access and manipulate it, (d) to support resolution encryption, (e) to support some ability to restrict access to values in the PID records. The FDO Forum together with related initiatives will work to ensure that these requirements are fulfilled. For details on PID Systems see the appendix.

[7] This implies F2: providing rich metadata, I3: qualified cross-references, and R1: relevant attributes.
[8] The term PID stands for global, unique, resolvable and persistent identifiers.

2.3 The FDO Record

The FDO record is the core data structure at the heart of the FDO model and key to enable machine readability, interpretability and actionability. The FDO record consists of a set of typed value pairs that includes the reference to the FDO's profile, its type, its metadata and data. These specific attributes will be defined in the FDO specifications and will themselves be FDOs. The metadata and data may be accessible in-line or by reference. Machines reading an FDO record will process it to determine the type of the FDO and what steps they should take next based on their contexts and goals and their decision may then be to access the FDO's metadata to fully characterise the data, or access the data, or following a found reference to another set of FDO records to recurse on.

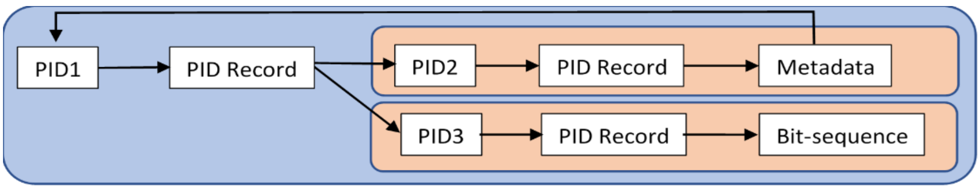
PID Records are accessed using FDO access protocols to abstract from the specific technology used to store it. It is important to note that a PID Record can be itself a FDO Record.

2.4 FDO Configuration Variants

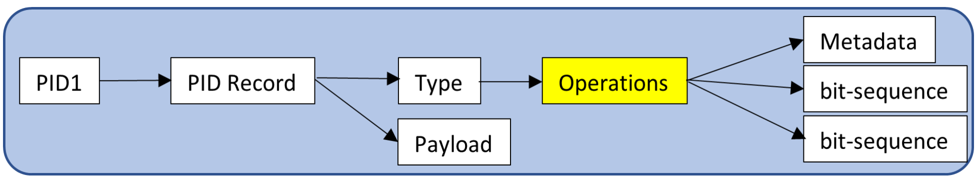
As shown in the "Configuration Types" document [FR3], FDOs can exist in many different variants while remaining FDOs, fulfilling all requirements. The ConfigurationTypes Document concludes the following:

There are several different ways to organise FAIR Digital Objects (FDOs) and nevertheless adhere to the FDO Requirements. In all definitions of the term "FAIR Digital Object (FDO)" this variety needs to be considered.

Users will request more guidance in terms of, e.g., implementing FDO systems. Therefore, we are introducing the term "Recommended FDO Configuration Types" and assign it to two organisational models[9][10]*:*



- *The first recommended model offers a high granularity in the case of closely bundled objects. Every component is represented as its own FDO and is aggregated in an overall FDO. This approach ensures that all components can be addressed separately.*



[9] The diagrams still use the old terminology of „PID Record" which in this context will be replaced by the more adequate term „FDO Record".

[10] In these simple diagrams the arrows mean the following: PID – *resolves_to* - PID record, PID record – *contains* – references, metadata – *contains* – references, Type – *enables_association_with* - operation

- *The second recommended model offers a high flexibility in adding a variety of different components all linked together by defined operations on information types.*

It should be noted that there are FDOs need which do not contain a structured bit-sequence but could include just a set of references as indicated in the first example for PID1. In addition, FDOs could contain references to queries, for example, that are executed on tables to return the expected bit-sequence.

It is also possible to create other configuration variants to accommodate other use cases. Some variants could consist of a PID resolving to a PID record that itself is an FDO Record complete with the metadata and data. Other variants could be more complex and accommodate further nesting or aggregation models based on the FDO Record.

2.5 FDO Profiles, Kernel Attributes and Metadata

Two early documents [FR5, FR6] describe the nature of PID profiles[11], kernel attributes and make statements about metadata. These have been refined by documents about the FDO Typing system.

The following recommendations are made:

1. Each PID registration must be associated with a PID Profile and the reference to the profile must be determined by a mandatory attribute in the PID Record.
2. The PID Profile is an FDO and must have a specified and unified structure.
3. The attributes being specified in the PID Profile and instantiated in the PID Records need to be registered and defined in open and recognised registries.
4. The PID attribute set should be limited to properties which are required for quick inspections and immediate processing and contains

1. **mandatory** attributes specified by the FDO Forum
1. **optional** attributes specified by the FDO Forum some of which are **recommended** and
2. **community** defined mandatory and optional attributes specified by communities

Mandatory Kernel Attributes

- “Reference to the PID Profile” allowing machines to find the definitions of the attributes being used by the service provider who generated the PID Profile
- “Type of the FDO” allowing machines to associate operations with FDO’s bit-sequence

Optional Attributes

To increase interoperability, the FDO Forum will define and register a set of attributes that will be frequently used across sectors/disciplines. PID creators should use these attributes as much as possible but are free in their choices. A process needs to define how to adapt the set of attributes over time.

Recommended Optional Attributes

A set of recommended attributes are thought necessary for operating and using FDO’s in practice. Nevertheless, they are optional.

Community Provided Attributes

[11] The term PID Profile is synonym to the term FDO Profile. In the context of FDOs the latter is seen as more suitable.

Many communities have well defined metadata schemas that contain information about attributes which are being used broadly. Such communities will want to make use of attributes they defined and registered properly also in the context of PIDs. Specifications will have to be made that specifies requirements with respect to the type of registries that can be integrated.

Community Provided Metadata

In many cases community-provided metadata includes many references or information that is relevant at FDO management level. As indicated above, FDO Forum will specify mandatory PID attributes, i.e., the FDO Forum requires that the corresponding attributes need to be extracted from available sources and filled in into the PID record.

2.6 Typing System

2.6.1 Principles on Types [FR8]

The key reason for assigning Types to FDOs is to enable machine actionability, i.e., giving compatible clients the information they need to act on FDOs. An FDO may have any sort of data and metadata, but each must be typed using a type-value pair.

Every FDO will have an FDO Type enabling machines to quickly determine what further actions are possible. FDOs can use any existing type or define new types, either completely new or derived from existing types. Attributes can be used as a method to attach types to FDOs. The association of an FDO with the FDO Type is specified by a first class attribute (called Type-Key) in the profile or the PID Record of the FDO.

An FDO is not considered complete unless its type is defined and registered. It must be possible to resolve a type identifier to a corresponding registered type definition FDO.

Type definitions

The type of an FDO will be specified using an agreed upon syntactic description that may be structured and dependent on components that are themselves FDOs. This syntactic description will be called an FDO Type definition in the following and is an FDO itself.

A set of agreed-upon (mandatory and optional) elements for expected common elements in type structure, e.g., dependencies on other types, etc., will be part of the FDO Type definition as a standardised structure. Every FDO Type and type is associated with a PID which resolves to the FDO that includes the type definition. So it is of type FDO type definition.

The FDO Type Framework

The FDO Type Framework (FDO-TF) [FR8] will specify methods for defining types for and attaching types to FDOs. The FDO-TF will define the manner in which FDO Types and new types created within the FDO space are identified, described, and used. The FDO-TF will make use of existing type definitions that are openly available such that the FDO Types can reference relevant existing types, and thus leverage the work of standards, communities and best practices of all sorts.

A classification of FDO Types can contain categories like data, operations, FDO definition, FDO Type, FDO administration and other types of use to FDOs.

Type Registration

As FDOs will be applied across multiple dynamic disciplines and sectors, there can be no single ontology governing all types. There will be no centrally controlled type registries, but communities are expected to endorse types or collections of types, and a central index or search service for types is highly desirable.

The FDO communityForum will propose a governance structure for recognizing community endorsed collections and search services for FDO Types.

Operations and Relations with Types

Specifications of types of data and operations that can work on such types need to be independently managed, as a variety of operations on types and the set of possible operations will change depending on the purposes.

The set of operations possible for FDOs with a given type is not a predefined property of a type, but types can provide necessary conditions for the applicability of operations. There must be an approach to defining the relations between and among types.

2.6.2 Implementation of Attributes, Types, Profiles and Registries [FR9]

The general concept of an FDO consists of three layers:

- the PID System,
- the FDO model layer that defines the structures and semantics needed to implement machine actional binding and
- the resource layer managing the different resources in a distributed landscape of trustworthy and FAIR compliant service providers.

Community metadata and the bit sequence attached to the PID comprise the major components of the FDO and the attachment needs to be made by references given by attributes. The bit sequence is the usual reference provided by current PID systems in a generic resolution process. A machine actionable binding for community metadata, however, is only provided by community specific approaches and there is currently no generic way for machines to operate on and with that metadata, for instance to understand the inner structure of the bit stream. The FDO approach is an attempt to overcome this lack of automation by providing a framework with machine actionability by design.

Machine actionability requires complete syntactic explicitness which can be provided for metadata in general by a model of keys and values, called attributes, where the key itself is a reference to its syntax description and the syntax of the value follows the requirements of the referred syntax.

Therefore, such syntax descriptions require a generic access mechanism which can be given again by a PID. They need to be machine readable, so they need to be FDOs themselves. Also they need services which maintain the attribute definitions, called attribute registries.

A machine that works on an FDO also needs the information on which attributes it can expect, which is additional metadata that comes with a special attribute called profile. Also the syntax of profiles needs to be defined and there must be an agreement on mandatory attributes. Furthermore, the profiles themselves are FDOs and need services which maintain the profiles, called profile registries.

The document on Implementation of Attributes, Types, Profiles and Registries describes in more detail these and other related components and the relations among them. It describes some possible policies regarding these components and gives suggestions about the governance of these components. It also gives guidelines on how to implement these components without stating explicit technology or a particular implementation framework. It provides also examples of existing implementations of the described components.

3. Illustrations and Endorsements

The FOD Forum also worked on a few documents that illustrate working with FDOs and it endorsed external specifications. They are summarised here.

3.1 FDO Uploading

The document “FDO-Upload” [FR10] is not normative in the sense of adding specifications to the Full FDO Specification.

It includes an illustration about the kinds of procedures that are involved when a new FDO or a set of new FDOs will be uploaded to the known domain of FDOs. The FDO Forum needs to offer a set of recipes for different cases, the sketched Upload Cases point in the direction of such recipes.

In addition this document indicates the layers at which services can be provided and how metadata elements provided by the community, which are crucial for FDO level processing, could be integrated into the FDO world.

3.2 FDO Granularity, Versioning and Mutability

The document” Granularity” [FR7] makes the following recommendations:

- *The granularity choices for FDOs are much aligned with what has been written so far on the choices for granularity for the assignment of PIDs, which allows the FDO Forum to refer to existing recommendations that emerged in the realm of GEDE and RDA discussions.*
- *The granularity for building FDOs is dependent on pragmatic utility decisions within the communities of practice: what are useful entities to work with in the corresponding application field. Early choices can, with more or less difficulty depending on the specifics, be revised by (1) breaking up the bit-sequences of a given FDO into parts and creating new FDOs assigned to the parts or (2) by aggregating individual FDOs into collections which are also FDOs.*

- *Data providers need to make clear policy statements on whether the content of an FDO is mutable, i.e., can change over time such as is often the case in databases, or is immutable. Inspecting the accuracy of these statements should be part of any auditing procedures. Therefore, the addition of a 'mutability' attribute into the PID record is recommended.*
- *There are different policies between communities on how to indicate versioning – some use the PID string, some use the PID record and some use extended metadata for these purposes. The choice depends on efficiency criteria and no general recommendation can be made except that data providers need to make clear statements on how versioning is handled.*

The specifications about granularity, versioning, and mutability of data is thus not an issue that need to be defined by FDO specifications. FDOs are transparent to these decisions. It should be good practice, however, for the communities of practice to make clear specifications about their policies in these respects. In addition, FDOs should provide optional kernel attributes that allow communities to add information about versioning and mutability in a cross-disciplinary way.

3.3 DOIP V2.0 Endorsement

The DOIP Version 2.0 as released by the DONA Foundation [FR11] is endorsed as an important contribution to the FDO Forum specification stack. The DOIP interfacing protocol specifies how clients can interact with DOs independent of the technology used by a DOIP server and independent of the data organisation chosen. Since FD0s are a subset of DOs, restricted by the requirement to be FAIR compliant and thus machine actionable, DOIP will also work for FDOs as specified by the FDO Forum.

DOIP is an important component for implementation of FDOs using the DO approach. It has been tested by several institutions and will be subject of further intensive tests within the realm of FDOs.

Appendix: PID System

Globally unique and resolvable persistent identifiers (PIDs) are at the core of the FAIR Digital Objects, which is why we describe in some details the key PID requirements that are essential for FDOs in this specification document. For some basic definitions this document refers to the RDA DFT Data Core Model[12] which we have refined to serve for FDOs.

1.1 Persistent Identifier (PID)

A persistent identifier is an ID represented as a string that uniquely identifies an FDO with the intent to provide persistent and consistent resolution over time to meaningful state information about the identified FDO. The term "PID" includes a scheme for identifiers and a global system that resolves them to useful information.

1.2 PID Record

A PID record is the structured information returned from a PID resolution. It consists of a set of attribute (key) value pairs. The FDO PID record may contain any user specified attributes but shall have the mandatory set of attributes as specified in the FDO specification.

1.3 PID Service

A PID Service is a publicly available global infrastructure of services that manages the association of PIDs to PID Records and has a public protocol that can be used by any client around the world to securely resolve PIDs into their associated PID records. The administration of the PID service is outside the scope of the PID Service FDO requirements.

1.4 PID Resolver

A PID resolver is a piece of client software that can be used to connect to a PID Service, combining to create a globally available infrastructure system that has the capability to resolve a PID into useful, current state information describing the properties of an FDO.

For more information on PIDs we refer to the report "A Persistent Identifier (PID) policy for the European Open Science Cloud".[13]

3.1 A Persistent Identifier that supports and enables research that is FAIR is one that is globally unique, persistent, and resolvable.

With what is called state information in the RDA DFT Core model [], the EOSC PID Policy documents [ref] speak about Kernel Attributes, a term that has been detailed in the RDA Kernel Information Type group and which we will use for FDO purposes.

Kernel Attributes: A global resolution system should resolve any PID into a set of Kernel Attributes describing properties of the object it identifies. Kernel Attributes are contained in a structured record with attributes whose semantics are retrievable in machine-interpretable form. In general, the set of Kernel Attributes should at least contain attributes that point to where the bit sequence

[12] http://hdl.handle.net/11304/5d760a3e-991d-11e5-9bb4-2b0aad496318

[13] https://op.europa.eu/o/opportal-service/download-handler?identifier=35c5ca10-1417-11eb-b57e-01aa75ed71a1&format=pdf&language=en&productionSystem=cellar&part=

of the referent can be found and a type definition. Optionally, it may contain pointers to further contextual objects including metadata. PID Profiles which define the set of Kernel Attributes used by a specific FDO service provider should be registered in open registries.

3.6.1 Nature of PIDs and PID Systems

Many solutions for identifiers exist currently. In this paper we focus on identifiers which are useful to identify digital entities of various types and that have shown robustness for some decades already. We will ignore identifier systems that do not provide a resolution mechanism to access property information about the digital entity they identify even if they fulfil important functions such as UUIDs, for example. We also ignore all the many identifier schemes that are being used locally in data systems (database management systems, clouds, etc.). Finally, a variety of communities have defined their own PID system, but they are meant to address particular problems and we do not address those here.

When looking for persistent identifiers (PIDs) systems that satisfy our requirements we will restrict ourselves, in this document, to URIs (URLs, URNs) and Handles/DOIs. Since URI/URLs are widely known and Handles not, we describe the latter in more detail.

URIs

URIs are defined by W3C and occur in two versions: URLs identify, which locations (Uniform Resource Locators) in the web and URNs (Uniform Resource Names), which identify names. A Uniform Resource Locator (URL), colloquially termed a web address, is a reference to a web resource that specifies its location on a computer network and a mechanism for retrieving it. A Uniform Resource Name (URN) is a Uniform Resource Identifier (URI) that uses the urn scheme. URNs are globally unique persistent identifiers assigned within defined namespaces so they will be available for a long period of time, even after the resource which they identify ceases to exist or becomes unavailable.

URLs, as currently used, mix identification and location and due to this mixing are not stable or persistent. Various constructions such as Persistent URLs (PURLs) have been invented to bypass the instability problem, i.e., a PURL is registered and points to the web-resource using the indirection feature. If a resource is being replaced the PURL needs to be changed. URLs are used world-wide.

URNs were invented to overcome the weakness of URLs by defining name spaces which have the authority to assign persistent names to resources. URNs are using a different scheme, are not popular, and despite several initiatives there is no global resolution system.

Handles/DOIs

The Handle Scheme is based on RFCs and a resolution system is being provided in two layers: (1) A Global Handle Resolver which is distributed and redundant and which is taken care of by the independent non-profit DONA Foundation (Geneva). (2) Prefixes are assigned to Local Handle Resolvers which define name spaces in this distributed PID landscape where responsibilities and accountability is clarified by legal documents and persistence needs to be granted by community agreements. Each local Handle service provider needs to define a business model, procedures according to which it will act and make implementation choices that guarantee stability, robustness and security. An example of such a local Handle provider is the International DOI Foundation issuing DOIs.

3.6.2 Identifier Resolution Protocols

In accordance with the FAIR principles the protocols used to resolve a PID into useful information need to be standardised, freely accessible and publicly usable. In the case of URI/URLs the well-known HTTP protocol will yield a HTML based landing page which can include any kind of information and thus is not restricted.

In the case of Handles the resolution protocol was defined by RFCs which are now being replaced by the DO-IRP protocol, which is compliant with ITU X.1255. According to these specifications the resolution system returns a package of attributes (key - value pairs), and the package contains structured information which, when the attributes are defined and registered, can be interpreted by machine or human users.

3.6.3 Binding in PID Systems

As has been indicated in previous sections of this document a crucial requirement of the FDO is, the capability of the PID of an FDO to be bound in a standard and consistent manner to the set of attributes describing that object to provide for consistent resolution of a PID. In the case of URI/URLs, there are many possible ways to do this, including requesting an HTTP HEAD request on a URL and processing a MIME header response, or performing an HTTP GET request on a URL and parsing the landing page for relevant information. The HTML landing pages could be structured to provide a standard set of fields as has been worked out by the Signposting suggestion where type semantics defined in the IANA registry or types minted by the community according to a community process. are expected. The Signposting approach does have a few restrictions such as no possibility to include specific metadata attributes for data integrity checks such as checksum and X.509 certificates to prove authenticity, limitation to the set of types that can be included, and the need to parse MIME header syntax to identify attribute-value pairs. The limitation and difficulty of any of these URI based solutions rests in the fact that, given a URI, a client has no ability to determine which solution it can use to acquire information about the object identified by the URI other than trying all the methods it knows about.

The Handle System, which is an implementation of DO-IRPV3.0 (DO-IRPV3.0 replaces the Handle RFCs 3650, 3651, 3652 and is backwards compatible) implements the standard's resolution protocol allowing any client to resolve a handle into a standards defined handle record. This standardisation of the resolution response is a great asset to all processes that interact with handles or DO-IRP compatible implementations.

The Handle System defines a few system wide types to support its operations but leaves to communities of practice to specify the specific type of attributes to include in the handle record to achieve the interoperability desired. This feature enables the FDO community to create its own profiles and Kernel Attributes. One critical attribute is the type of the FDO which assists the clients in quickly determining the nature of the FDO and what operations it can perform on it. This approach offers a clear path towards encapsulation and automation of FDO processing.

3.6.4 PID Security

As the PID System is at the core of the FDO security and at the base of the trust pyramid it is important to describe some of its relevant properties. Access to HTML pages can be secured by the usual web-mechanisms, i.e. access can be restricted using access control lists maintained at the operating system level of the resource provider. Also encryption of the HTML page is possible.

The Handle System offers the following security/privacy options: (1) The PID creator is the "owner" of a PID record, not the PID system provider, and only the owner of a PID is allowed to manage the content of the PID record. (2) The owner of a PID can authenticate using a public key system using standard X509 certificates if necessary. (3) The owner can restrict the PID record at the attribute-value pair level, i.e. to restrict access on certain critical attributes. (4) The PID System can provide encryption and non-repudiation of its values. (4) The PID owner can select from a range of PID service providers it trusts.

3.6.5 PID Scalability & Robustness

In the evolving data processing domain, given the increasing need for automated processes, scalability and robustness of the PID system will be key. The URI/URL and its DNS/HTTP based resolution service has shown its robustness and scalability in resolving URLs to HTML pages and resources. While the performance of the DNS/HTTP resolution solution is more than appropriate for a page based access, the lack of a consistent approach for returning interpretable information makes this solution unsatisfactory when wanting to provide resolution capabilities to very large large numbers of PIDs across a wide range of domains, communities, and services.

The Handle System was designed from the ground up to provide a high-speed resolution of Handles (PIDs) globally across many distributed and independent handle services around the globe. As such it is particularly well suited to scalably and to efficiently resolve many billions of FDOs into FDO records globally. The Handle System specifically implements a service architecture that enables a wide range of scalability solutions from large scale mirroring, caching services, and record caching that together can be used to guarantee resolution scalability. At the global resolution level currently 10 globally distributed rootnodes (called Multi Primary Administrators) provide a highly scalable and robust service. At the local resolution level communities can make the systems as scalable and robust as is needed. The example of the DOI resolution has demonstrated this scalability and robustness over two decades