FAIR DIGITAL OBJECTS FORUM

# FAIR Digital Objects
# Full Specification

# Version PR 1.1

FDO Forum Proposed Recommendation 17. October 2022

**Current and previous versions:**

Current version:

**Editors:**

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

**Authors:**

I. Anders, Ch. Blanchi, Daan Broder, 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 the first comprehensive version of a specification for 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 with many examples and illustrations and in a verbose style allowing many to follow the discussion. This document will in most cases only collect the essentials of these documents and integrate them into one summary document on FDOs.

It will also draw a general view on FDOs as far as can be extracted based on the specifications so far.

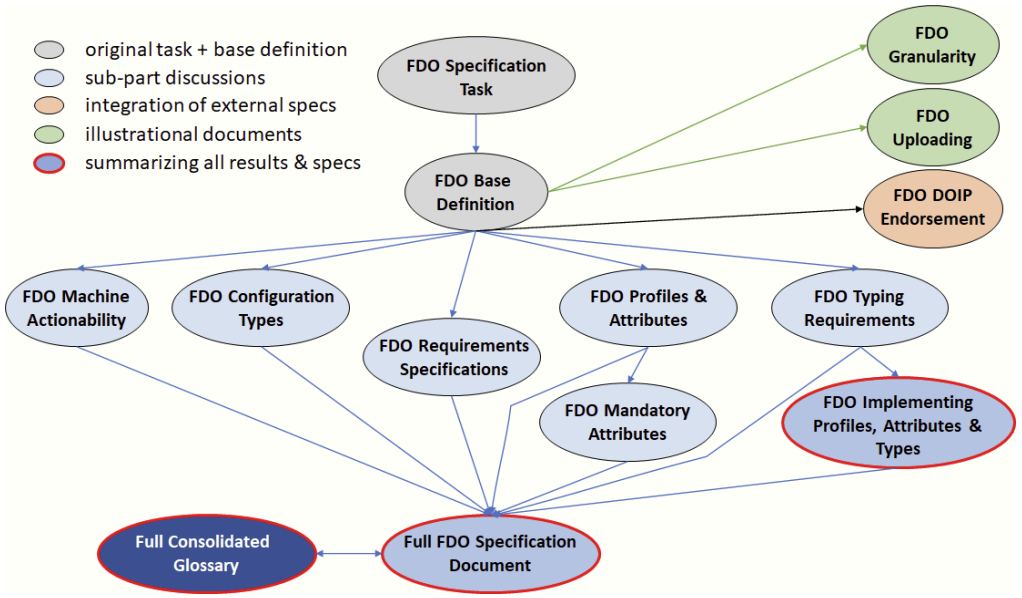
The complete list of FDO specifications to date are at the end of this document.

# Status of this document

This document version is a Proposed Recommendation 1.0 that can be discussed broadly, since it is based on documents that are in a proposed recommendation state..

# How to read this Document

**Section 1** contains easy to read descriptions of what FDOs are. After having agreed on a Base Definition (**section 2,** also containing some other basics), the WGs started to decompose the task of




specifying FDOs in detail into sub-tasks. Six documents with effects on the final specifications were discussed according to the FDOF Process document, two documents have illustrational relevance and one document contains an endorsement of an external specification relevant for FDOs.

**Section 3** includes relevant summaries of these documents and an elaboration on PID systems..

**Section 4** contains the aggregated specifications for FDO from the various documents. Since the FDO Implementing Profiles, Attributes & Types document is detailed it can also be seen as a final specification document. These two light blue documents with red margin are in a final discussion state.

The urgent next step is to write a **consolidated glossary** as a separate document to unify the terminology within the FDOF realm.

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:

- FDO Base Definition:
- FDO Requirement Specifications:
- FDO Configuration Types:
- FDO Machine Actionability:
- FDO PID Profiles and Attributes:
- Mandatory and and Optional Kernel Attributes:
- FDO Granularity, Versioning and Mutability:
- FDO Typing:
- FDO Implementing Attributes, Profiles, Types
- FDO Upload Document:
- FDO DOIP Endorsement:

This document also refers to a number of external specifications that have been endorsed by FDO or that are referenced by some of the above documents:

- DO-IRP
- Handle System Specification
- RDA Data Type Registry
- EOSC PID Document
- DOI Specifications
- RDA Data Foundation and Terminology
- RDA Kernel Type Information
- RDA Research Collections
- Research Objects and RO-Crate
- Signposting Specifications
- Luiz Bonino's Web-page
- CoreTrustSeal

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 need to thank all WG members for their contributions.

# Contents

# 1. General View on FDOs

## 1.1 Overview

FAIR Digital Objects (FDO) are uniquely identified distributed entities accessible on the Internet. They are abstract, technology, sector/discipline neutral, and autonomous units of information that can be used to persistently bind all necessary information about any digital entity so that it can be processed by humans and machines and therefore enabling the FAIR principles to be turned into reality.

FDOs are based on extending the generic core data model as described by RDA for Digital Objects, bundling necessary information such as metadata, rights and usage information in a way that is independent of technological choices and sector/discipline specifics. FDOs will facilitate their access by humans and machines for the long term while simultaneously allowing to enforce its access control, enabling data sovereignty fostering increasing trust. As such, FDOs will improve the potential to manage human's digital domain for centuries, to help to avoid a dark digital age and tackle the UN's Sustainable Development Goals.

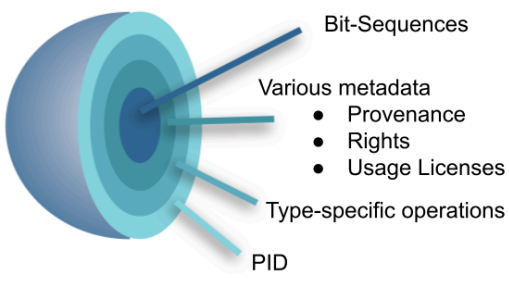
***There is an increasing pressure to design and develop a global integrated data space that fosters automated machine access and processing of data to benefit the society and economy. FDOs, thanks to their autonomous and technology independent characteristics, will be at the core of it. Thanks to their inherent FAIRness, they will also foster automatic processing.***

## 1.2 Description

The FAIR Digital Objects are at the core of the evolving global integrated data space in the same manner as as was the Datagram data model with the corresponding TCP/IP protocols the Internet. The FDO consists of the following.

(1) A unified core data model enabling a basic protocol to interact with it. This provides independence from the underlying way data service providers, such as repositories, implement, organise and manage their collections. The FDO specification promise to greatly foster interoperability across service providers thanks to its flexibility and its ease of implementation. The connectivity complexity will be reduced from $m^2$ to m.

(2) AAn FDO consists of three layers: (a) the PID System, which resolves a global unique resolvable and persistent identifier (PID[1]) into a consistent PID record binding the necessary information, (b) the FDO data model layer that defines the structures and semantics needed to implement machine actionable binding and (c) the resource layer that manages the different resources in a distributed landscape of trusted and FAIR



**FDO**

sed in so many documents.

compliant service providers. The PID System and the FDO model layer need to be exactly specified syntactically and semantically by the FDO Forum to guarantee predictable behaviour and machine actionability. The Resource layer will be managed by communities and practices taking care of FAIRness.

(3) A wide range for 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 associated with the PIDs.

(4) A PID system that is global, robust, scalable, persistent, secure, and resistant to attacks is at the core of the FDO's trust pyramid. The information associated with the PIDs needs (a) to be specified by an FDO Profile, (b) contains 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.

(5) FDO 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 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 Perspective

Today the data space is huge, heterogeneous and fragmented leading to 80% waste of time in data-driven projects in research and industry and this issue will only grow over time. The massive and ubiquitous deployment of smart Internet-of-Thing devices will lead to an even larger increase of the data space and its inherent complexity and even more smart developers will create technical solutions addressing certain aspects more efficiently. Also new technologies such as Quantum computing and DNA storage will become available with dramatic consequences for technological solutions.

It is now time to define the bases for convergence since the costs of fragmentation are already too steep and the increasing complexity hampers innovation and wide participation. We are at the bottom of a steep hill of transformations pointing to a flourishing digital domain and cannot yet see the kind of dynamic evolutions and disruptions that will occur. FAIR Digital Objects, as simple and autonomous persistent units, however, have the capacity to survive as a basic core model for data in the rough sea of technological innovation and thus to act as an island of stability for the coming decades if not centuries.

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 the DIN (German Institution for Standards) to turn all specifications to international standards in the realm of ISO.

# 2. Basics

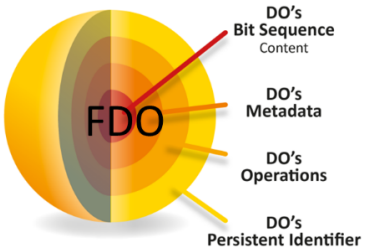
## 2.1 Base Definition

The FDO Forum agreed on the following base and a slightly extended technical definition of a FAIR Digital Object. They describe the canonical FDO configuration where the FDO includes references to bit-sequences encoding some relevant content of any type (data, metadata, software, semantic assertions, etc.).

**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 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.*

## 2.2 Core Data Model




The core data model of the FAIR Digital Object as schematically sketched in the diagram supports

- the highest form of abstraction in so far as it can include any type of content encoded in the structured bit-sequence such as data, metadata, software, semantic assertions, etc.
- a persistent binding of all relevant information (metadata) that is necessary to find, access, interpret and reuse (FAIR) with the bit sequence
- encapsulation in so far as each FDO has a type which can be related to all kinds of operations working on the bit sequence.

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 way when a given PID is being resolved.

In general, it is assumed that resources of FDOs are being 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 made such as by CoreTrustSeal which resulted from collaborations in the Research Data Alliance..

## 2.3 Technology and Implementation Independence

The specifications of the FDOs enable different implementation approaches and are transparent to the technologies being used to implement its components. As examples for implementation approaches we can refer (1) to solutions using the Web specifications with URI/URLs as PIDs, HTTP as interaction protocol and HTML and/or Linked Data as core data model or (2) to solutions using Handles/DOIs as PIDs, DO-IRP and DOIP as interaction protocols and Digital Objects as core data model.

## 2.4 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 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 Object was then also used in database systems which got 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 vaste amounts of files efficiently. Here each digital entity stored in the cloud has assigned a local identifier which gives access to administrative information (metadata) about the entity including a reference to its location. In 2014 the RDA Data Foundation & Terminology working group used Kahn & Wilensky’s model to test it against a wide variety of practical cases and extended it to describe a core data model emphasising the principles of abstraction and binding. 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) and DO Identifier Resolution Protocol (DO-IRPV3.0) .

In 2019 experts who developed the FAIR principles and 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 active to close the gap to data management. Recently, the Signposting solution 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 high attraction 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).

# 3. Relevant Aspects

For understanding FDOs it is relevant to first describe a set of aspects that are related with the concept and that were described in the different official documents on FDO. The essentials are reported in this section.

## 3.1 Machine Actionability

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

1. There are yet no clear conceptual distinctions and widely agreed definitions about the terms “machine readability”, “machine interpretability” and “machine actionability”.

2. For FDO Forum purposes 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 3 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[2] are machine actionable if they are embedded in an FDO infrastructure that implements policies, rules and procedures for FAIR?

[2] This implies F2: providing rich metadata, I3: qualified cross-references, and R1: relevant attributes.

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.

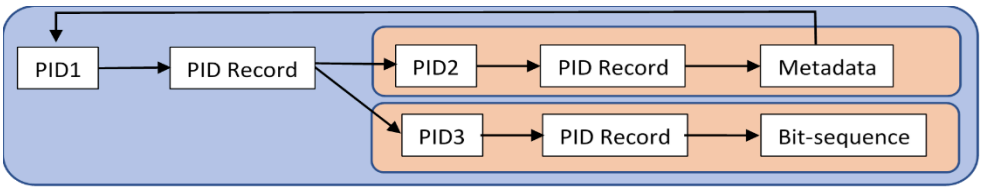
## 3.2 FDO Configuration Types

As is demonstrated in "ConfigurationTypes" FDO can occur in many different variants but still being 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 Framework Requirements. In all definitions of the term "FAIR Digital Object (FDO)" this variety needs to be considered.*

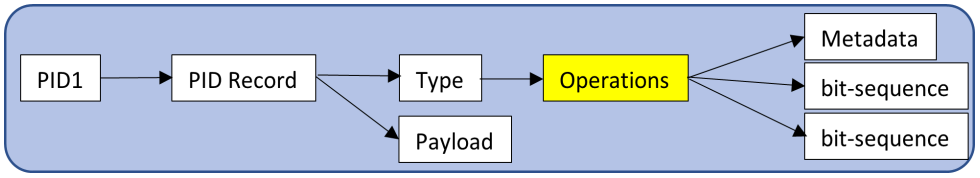
- *Users request more guidance in terms of, e.g., implementing FDO systems. Therefore, we are introducing the term "Recommended FDO Configuration Types" and assign it to two organisational models.*

    - *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.*

    - *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 which do not contain a structured bit-sequence, but 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 being executed on tables to return the expected bit-sequence.

# 3.3 FDO Granularity, Versioning and Mutability

The document" Granularity" 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 community of practices: what are useful entities to work with in the corresponding application field. Early choices can always be revised by (1) breaking up the bit-sequences of a given FDO in 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 whether the content of an FDO is mutable, can change over time such as often in databases) or immutable. Inspecting the correctness of these statements should be part of auditing procedures. Therefore, it is suggested to add an attribute "mutability" into the PID record.*
- *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 widely 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, handling of 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, that the communities of practices make clear specifications about their policies in this respect. In addition, FDOs should provide optional kernel attributes that allow communities to add information about versioning and mutability in a cross-disciplinary way.

# 3.4 Typing System

## 3.4.1 Principles on Types

1. The key reason for assigning types to FDOs is to enable machine actionability, i.e., giving compatible clients the information they would need to act on FDOs.
2. Every FDO will have an FDO type enabling machines to quickly determine its purpose.
3. An FDO may have any sort of data and metadata but each must be typed using a type-value pair.
4. FDOs can use any existing type or define new types, either completely new or derived from existing types.
5. The type of an FDO will be specified using an agreed upon syntactic description that may be structured and dependent on components that are FDOs again. This syntactic description will be called FDO type definition in the following and is an FDO itself.
6. 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 type structure.

7. 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.
8. 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.
9. 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.
10. The FDO Type Framework (FDO-TF) will specify methods for defining types for and attaching types to FDOs.
11. Attributes can be used as a method to attach types to FDOs.
12. A classification of FDO Types can contain categories like data, operations, FDO definition, FDO type, FDO administration and other types of use to FDOs.
13. The FDO Type Framework (FDO-TF) will define the manner in which FDO Types and new types created within the FDO space are identified, described, and used.
14. 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.
15. 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.
16. The FDO Forum will propose a governance structure for recognizing community endorsed collections and search services for FDO Types.
17. As FDOs will be applied across multiple dynamic disciplines and sectors, there can be no single ontology governing all types.
18. 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.
19. 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.
20. There must be an approach to defining the relations between and among types.

### 3.4.2 Implementation of Attributes, Types, Profiles and Registries

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.

In this document the components of the FDO model layer are described in more detail and together with the relations between them that are needed for the implementation.

Community metadata and the bit sequence attached to the PID build 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 actional 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 places where the attribute definitions are maintained, called attribute registries.

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

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

## 3.5 FDO Uploading

The Upload Document "FDO-Upload-x.x" 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 thought of and how metadata elements provided by the community and crucial for FDO level processing could be integrated into the FDO world.

## 3.6 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[3] which we 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 persistently and provide consistent resolution over time to meaningful state information about the identified FDO. The term "PID" includes a scheme for identifiers and a system that resolves them to useful information.***

***1.2 PID Record***

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

***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 globally publically available 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 software that can be used by any client or service to contact a PID Service and system is 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".[4]

***3.1 For the purposes of this Policy, 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 speak about Kernel Attributes, a term that has been invented 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 about 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 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 ignore all the many identifier schemes that are being used locally in data systems (database management systems, clouds, etc.) 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. Also a variety of communities have defined their own PID system, but they are meant to address particular problems.

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

**URIs**

---

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

URIs are defined by W3C and occur in two versions: URLs identify locations (Uniform Resource Locators) in the web and URNs (Uniform Resource Names) 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 are thus mixing identification and location and due to this mixing are not stable and persistent. Various constructions such as Persistent URLs (PURLs) have been invented to bypass the instability problem, i.e. a PURL is being 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 distributed 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 to resolve a PID into useful information needs 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, and is compliant with ITU X.1255. According to these specifications the resolution system returns a package of attribute (key) - value pairs, i.e. the package contains structured information which, when the attributes are defined and registered, can be interpreted by machines or 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 a 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: Requesting a HTTP HEAD request on a URL and processing a MIME header response. Performing an HTTP GET request on a URL and parsing the landing page for relevant information. These 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 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 should 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 needs to provide encryption and non-repudiation of its values. (4) The PID owner should be able to select from a range of PID service providers it trusts.

### 3.6.5 PID Scalability & Robustness

In the evolving data processing domain with 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 to resolve 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.

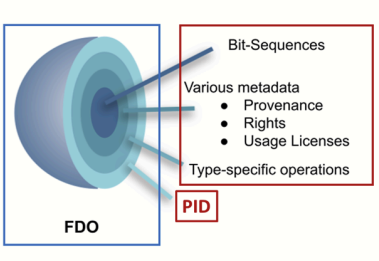
The Handle System was designed from the ground up to provide a high-speed resolution of Handles (PIDs) globally across many distributed and independent handle services around the globe. As such it is particularly well suited to scalably and to efficiently resolve many billions of FDOs into FDO records 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 (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 demonstrates scalability and robustness over two decades already.

# 4. FDO Specifications

This section summarises the requirement specifications for FDOs as they can be drawn from the various documents. As can be seen from the diagram an FDO includes 3 layers that need to be subject to specifications:

- the PID Layer
- the FDO model Layer
- the resource layer




We will start with the general requirement specifications as included in the FDO Requirement Specification document covering several layers and then describe specifications for each layer in more detailed ways where necessary. Obviously, the FDO requirements support the FAIR principles.

## 4.1 General FDO Requirement Specifications

**FDOR-GR1**: A *PID*, standing for a globally unique, persistent and resolvable identifier, is assumed to be the basis for FAIR Digital Objects. Every FDO is assigned one or more PIDs.

**FDOR-GR2**: A PID resolves to a structured record (*PID Record)* compliant with a specified *PID Profile* which leads to resolution results that enable programmatic resolution from PID back to the FDO and its elements as specified by these requirements. PID Records represent the information characterising FDOs and together with their resolving PIDs they can be themselves FDOs.

*Note: The term "PID Record" is synonymous to the term "FDO Record".*

**FDOR-GR3**: If an FDO contains a structured bit-sequence, the structured PID record includes at least a reference to the location(s) where the bit-sequence encoding the content of a FAIR-DO (FDO) can be accessed and the type definition of the FDO. The structured record may also contain PIDs pointing to *Metadata FDOs* describing properties of the target FDO.

**FDO-GR4:** The PID record needs to contain mandatory FDO attributes, may contain optional FDO attributes and attributes agreed upon by recognized communities. Values of attributes can be references. Expectations of such references should be specified in the PID profile or definition of said attribute in a Data Type Registry.

**FDO-GR5**: Each FDO identified by a PID can be accessed or operated on using an interface protocol by specifying the PID of a registered supported operation.

**FDO-GR6**: Any basic FDO interface protocol offers standard Create, Read, Update, Delete (CRUD) operations on FDOs and a possibility to use extended/domain operations for specific applications.

**FDO-GR7**: The relations between FDO Types and supported operations are maintained in separate registries.

**FDO-GR8**: Metadata can themselves be FDOs which describe the properties of the referenced FDO. They must be specified by a registered schema that refers to defined and registered metadata categories.

**FDO-GR9**: Metadata can be of different types such as descriptive, deep scientific, provenance, system, access permissions, transactions, etc.

**FDO-GR10**: Metadata schemas are maintained by communities of practice and are FDOs. Such metadata schemas should therefore themselves follow FAIR principles.

**FDO-GR11**: A collection of FDOs is also an FDO. The content of collection FDOs describes its construction using an agreed formal language which specifies the relationships of the constituent members. An FDO may be a member of several collections.

**FDO-GR12**: Deletion of an FDO must lead to standardised and thus machine interpretable tombstone notes in metadata and PID records. The PID itself is not deleted.

**FDO-GR13**: The PID resolution and the FDO Layer information must be “machine actionable” which are those elements in bit-sequences that are machine interpretable and belong to a type for which operations have been specified in symbolic grammar.

**FDO-GR14**: FAIR Digital Objects can be configured in different ways as long as the configurations are compliant with the FDO Specifications.

**FDO-GR15**: The granularity for building FDOs is dependent on pragmatic utility decisions within the community of practices: what are useful entities to work with in the corresponding application field. FDOs are transparent to these decisions.

## 4.2 PID Layer Specifications

**FDO-PIDR1**: In accordance with the FAIR principles the protocols to resolve a PID into useful information needs to be standardised, freely accessible and publicly usable.

**FDO-PIDR2**: The resolution of a PID needs to lead to predictable results which are specified by a machine actionable FDO profile.

**FDO-PIDR3**: The PID is resolved to a set of key attribute-value pairs defined by the FDO profile that needs to be machine actionable, i.e. all attributes need to be defined and registered in open registries compliant with the FDO or community standards.

**FDO-PIDR4:** The PID system which is used for the identification of FDOs must be global, robust, scalable, and demonstrate persistence.

**FDO-PIDR5:** The PID System which is used for the identification of FDOs must support high security capabilities. The owner of a created PID and its associated attribute-value pairs is the only actor to make changes, to define accessibility to attribute-value pair information, and to request encryption of information.

**FDO-PIDR6**: Access to the PID system needs to be secured by a public key infrastructure and if necessary the use of X509 certificates must be possible.

## 4.3 FDO Layer Specifications

This document does not include details about the schemas defining FDO profiles and attribute definitions. For this we refer to the document on **Implementation of Attributes, Types, Profiles and Registries**.

**FDO-FDOR1**: The content of each FDO-record must be described by an FDO profile in accordance with an FDO defined schema which is registered in a recognised registry.

**FDO-FDOR2**: The FDO record consists of a set of (kernel) attribute-value pairs as defined by the FDO Profile and all used attributes need to be defined and registered according to the type specification schema.

**FDO-FDOR3:** Each FDO record needs to contain the mandatory kernel attributes (reference to the FDO profile and the type of the FDO.

**FDO-FDOR4**: FDO records can include a) optional kernel attributes which are defined by the FDO Forum or b) can include other community defined and registered attributes if machine actionability is guaranteed.

**FDO-FDOR5**: In the canonical case that an FDO includes bit-sequences encoding FDO's content, the attribute-value set must contain the reference to these bit-sequences and a rich metadata description as requested by the FAIR principles.

**FDO-FDOR6**: Every FDO must have an FDO type enabling machines to quickly determine its purpose and use.

**FDO-FDOR7**: An FDO may have include any sort of bit-sequences (data, metadata, software, semantic assertions, etc.) but each must be typed using a type-value pair in compliance with the FAIR principles.

**FDO-FDOR8**: FDOs can use any existing type or define new types, either completely new or derived from existing types.

**FDO-FDO9:** The type of an FDO is specified using an agreed upon syntactic description that may be structured and dependent on components that are FDOs again. This syntactic description is called FDO type definition in the following and is an FDO itself and thus FAIR compliant.

**FDO-FDO10**: 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.

**FDO-FDO11**: The FDO Type Framework specifies methods for defining types for and attaching types to FDOs

## 4.4 Resource Layer Specifications

As has been mentioned, the resources referenced by the FDO are being managed by different service providers and we are confronted with much legacy. Therefore, we can only make general statements and expect much time being required until practices have been changed.

**FDO-RESR1**: All resources referenced by FDOs must be FAIR compliant.

**FDO-RESR2**: All resource and service providers should demonstrate trustworthiness by regularly assessing the quality of the procedures applied according to standards such as CoreTrustSeal.