RDA Data Fabric Configurations - Thoughts about PID Centric DMA: Towards a global Virtualisation Layer

Tobias Weigel, Michael Lautenschlager, Peter Wittenburg, Larry Lannom, Dieter van Uytvanck

May 9, 2016

This document describes a list of possible extensions of a PID Centric Data Management and Access Data Fabric configuration beyond the currently identified outputs from the PIT, DTR and DFT RDA groups and by using the Handle System.

# 1. PID Centric Data Management and Access (PID CDMA)

We agreed to form a group of interested experts within the realm of Data Fabric that will go beyond pure specification work and looking at testing RDA outputs individually, but will do evaluative testing **Core Components** (CoCos) in combination as evaluation is the precursor to adoption. Only this type of real-world testing will indicate whether the current specifications of current and future CoCos are sufficient or need to be changed, many such evaluations across communities will need to take place. Such combinations of CoCos can be seen as concrete configurations of data fabrics and there should be many different of them. In this document we focus on one such configuration which we call the “**PID Centric Data Management and Access**” configuration. It will be based on the reference software of the Persistent Identifier Types and Data Type Registry RDA recommendations and be in overall compliance with the model as suggested by Data Foundation and Terminology WG.

As is shown in the diagram below one basic assumption of PID CDMA is the existence of a trustworthy repositories[[1]](#footnote-1) that store digital objects[[2]](#footnote-2) as has been described by the DFT model, i.e. a PID can be used to fully specify how the bitstream of a digital object can be accessed be it in a cloud, a file system, a database or any other future storage system. Typically in PID CDMA frameworks actors, preferably software agents acting on behalf of users, deal with metadata and PID records that are used to store attributes about the digital objects. In particular PID records include what could be called system-metadata[[3]](#footnote-3) and binding information[[4]](#footnote-4). By talking about a PID CDMA we are in fact meaning that users deal to a large extent with representations of the Digital Objects, i.e. a virtualisation layer is being introduced which we could call the global Digital Object Cloud. Physical information (where it is stored, how it is stored, etc.) is hidden to the user which is a step which has already been introduced by cloud technology. PID CDMA means to introduce this kind of virtualisation at a level across all kind of stores, i.e. it is currently the most promising solution to revolutionise data management and access.



Experts committed to this model will work and collaborate on two strands of this configuration instantiation:

* work out a model for minimal information types associated with PIDs that works with the reference software of the Persistent Identifier Types and Data Type Registry RDA recommendations and do a concrete implementation in collaboration with communities
* work out more details of the proposed configuration and possible future directions

While the first strand focuses on the use of RDA outcomes with existing software, the second strand is looking for “easy to add” functionality to specify and add further components of this PID Centric Data Management and Access configuration. This document focuses on the second strand. An indication of possibilities is given in the drawing and below description:

Why undertake this? Minimal information types associated with a PID allow for rapid and lightweight accept/reject decisions to be made about the suitability of an object. PIDs resolve extremely quickly, so the minimal PID information could be used to make decisions about routing an object for instance. But the PID record could also contain the pointer to the full metadata description (see third diagram) so that richer decisions can be made. Pointers to the metadata embedded with the PID also ensure that the metadata record is not decoupled from the data record[[5]](#footnote-5) – it is bound forever through the metadata record associated with the PID. To enable this kind of functionality, the information in the PID record needs to have type information with it, and the definition of the type both registered and described in a way that both humans and machines can interpret the typed data. The PID thus serves, in essence, as a map or a guide to both locating a given data object, aka digital object, as well as interpreting and potentially reusing the object. As is indicated in the diagram a PID record contains minimal information about the Digital Object it references. This can be implemented as links that are “call by reference” or “call by value” in computer science terms. If the former is true, the PID record points to external reliable information sources. In either case the PID has a binding role. All of the pieces that go together to make a data object available and understandable over time have to be sustained, so they have to be chosen carefully. If this binding model is useful to many it can be used on which to add additional components.

# 2. PID CDMA Extensions

Components that could extend the basic PID CDMA configuration are centred on using the PID record as anchor point for information to be used by humans and for automatic processing. We assume that a DTR instance is ready to allow us to describe information types and that increasing amount of types will be registered. In the following a number of possible example components are briefly characterised.

## 2.1 PID Initiation Component

Crucial is the moment when a Digital Object is being registered with a (trustworthy) repository to become part of the open (searchable, findable, managed, accessible) data domain. According to the DFT model a DO is characterised by the existence of 3 related entities: bitstream, metadata, PID. The moment of registration is seen as the moment where PIDs will be assigned. We have a number of cases of registrations:

* some content (file, DB content, etc.) comes with metadata others not
* some content has already been assigned a PID according to some schema (and thus provider system)

Some content includes metadata in the header requiring an extraction of the metadata from the bitstream.

The registration API (Repository API WG) therefore needs to have at least the following attributes:

* path to the content
* path to the metadata description or header schema (such as for NetCDF files)
* existing PID

At the moment of the registration the following actions need to be carried out:

* a PID needs to be registered and added to system metadata[[6]](#footnote-6) (Metadata IG)
* a system metadata record is being created including relevant system information, the PID is added and the path to metadata is added in the PID record
* the PID record is setup to include references to system metadata, user metadata (Metadata IG)
* in case of header information metadata is being extracted and added to system and/or user metadata (header MD extraction could be called a separate CoCo which depends on the schema specification)
* metadata relevant for citation is added to the PID record according to the DataCite norms
* an empty provenance record is initiated compliant with W3C PROV and a reference to it added to the PID record
* dependent on further information to be registered (see below) more attributes are added to the PID record.

To implement this task we need agreements about some types, categories of system metadata and typical categories of user metadata.

## 2.2 DO Management Component (Move, Copy, Delete)

In this virtualised domain of registered digital objects[[7]](#footnote-7) we need to redefine operations such as copy, move and delete and urgently implement them, since they will be crucial for a functioning virtualisation layer in a global data domain. As the work in EUDAT has shown management of digital objects without such operations does not work efficiently. The traditional file management tools need to be replaced by a new set of tools.

Moving digital objects means

* to change the path to access its bitstream
* to update the PID record by the new repository which is not the owner of the PID record (see 2.3)
* to handle the access rights and licence issues somehow (see 2.4)
* to eventually change ownership of the PID record
* to eventually update its associated information (metadata object, landing page, etc.)

Copying digital objects means

* to move a DO (see above)
* to add an information in the PID record which repository has rights to change access rights, original metadata, etc.

Deleting digital objects

A typical problem is that certain bitstreams will not be necessary anymore after a certain period of time, mostly since new equipment will allow creating better observations. A component should be available to data managers or policy workflows to delete one or a set of bitstreams where the case that instances in other repositories will stay accessible needs to be taken care of in the component. Only if all stored instances have been deleted the “deletion” flag should be set. But the corresponding PID and metadata should be kept so that references still lead some useful information.

## 2.3 Remote Change Component

The PID record will get a key role in the PID CDMA domain making it a necessity to make information highly redundant and secure. Nevertheless, we have increasingly often the situation that DO will be exchanged between trusted partners in federations. These partners will need to change certain information such as the path to access copies of the bit streams. The owner of the PID record however, must be the repository where the DO is being registered to become part of the domain of accessible data, i.e. the owner must have the right to carry out modifications and this in a highly controlled way.

This conflict of permissions needs to be solved by a secure component that allows trusted partners to modify specific attributes and where consistency will be immediately checked. An efficient solution to this problem requires that it is stored somewhere which partner in a federation is allowed to change specific information. Some brainstorming needs to be done to think about how to solve this problem which currently requires very inefficient and complex solutions.

## 2.4 Rights DB Component

Authentication systems have been suggested and have been deployed since some time and there is much experience already for systems working in distributed environments. Solving the authorisation part is not tackled really for large trust federations. What is the problem?

* Data is being copied to various repositories for various important reasons. In most cases the access permissions should be the same for all copies. The authorisation records in most cases need to be managed due to a number of obligations by the “original repositories” where the data has been registered. The managers of this repository are also in charge of maintaining the authorization records or have a system to empower the depositors to set permissions. How can these authorization records which are very sensitive be kept synchronous between the various repositories that offer instances of the data? We suggest to think about a secure “authorisation database” solution for trusted federations where the managers of the “original repository” retain the rights to set permissions and to delegate those if wanted, but to which all repositories that hold copies have read access (or in case of delegation also write access). This means that there is a central secure place in a federation to which the local permission settings of the “original repository” will be uploaded and where trusted repositories have access.
* Another aspect is that repositories are using a variety of licenses to give access to their data. Often these licenses are almost identical to the licences being used by other repositories. Getting access often means that a) access permissions have to be set and b) a license has to be signed. A license server would solve this problem efficiently if machines checking rights could simply check whether the right license has already been signed by a given user. Currently a user needs to sign licences with every repository again and again. As far as we know only the Finish HAKA federation suggested and built such a license server based on a unification of license types.

In both cases a reference in the PID records to the right entries would help machines finding the correct license and permission information.

We suggest starting Working Groups on these two issues to work out broadly accepted specifications, since the current mechanisms do not work anymore. Wrt the license server we should closely collaborate with the RDA WG on Legal Interoperability.

## 2.5 Collection Builder

Scientists want to build collections by aggregating (virtually) digital objects from various locations. At the end scientists do not want to know where the DOs are being stored, they want to form a collection, document this collection and execute some processes on it and o/or cite it. Some communities who are working in the directions of a PID CDMA already have built first versions of collections builders (<https://www.clarin.eu/content/virtual-collections>) already which allow to specify the new collection and store the metadata.

Another example is from climate research. In the next generation of climate model inter-comparison projects (CMIP6) hundreds of millions of data objects have to be organised in the global data federation. PIDs will be used to organise the data and metadata objects and their collections (<https://docs.google.com/document/d/13VjI377yNRnBE9fHkAqRY5o-QlUQvlFaoTkPdwZaNio/edit>).

However, more needs to be done since finally the collections need to be executed on some machines. Dependent on the type of processing it may be necessary to copy data temporarily to a new location. This could be sorted out by the collection builder so that the scientists do not have to deal with details. If the collection has a PID the only information the researcher needs is the PID of the collection and the moment when everything is ready to be processed.

## 2.6 Consistency Check Components

There should be components that check the consistency of all entities and their availability with respect to the basic model, i.e. check whether all PIDs and path specifications are correct and whether the indicated resources are available and accessible and take measures to reconstruct information if something is missing or not correct.

## 2.7 DOI Creation Component

It is widely agreed that DOIs are being used for the citation of published DOs. Within data fabrics in the scientific labs however researchers will not register DOIs for all the millions PIDs which they are creating. At certain moments in the lifecycle of selected DOs (mostly collections) researchers may want to turn PIDs into DOIs or create a DOI in addition in general also after having ensured that data and metadata has a certain quality and that persistence of the bitstreams is guaranteed at least for a certain time period. A component should be available that allows users to define and/or select collections and to create DOIs.

## 2.8 Metadata Extraction Component(s)

A brokering component should be made available that can extract metadata information from various well-known formats including headers such as NetCDF, JPG, etc. into registered metadata schemas which are being used by the community.

A second step would be to automatically create from the existing metadata being referenced by pointers in the PID record the metadata that is being requested for citation purposes for example by DataCite. In fact one could replace the citation metadata within the PID record by a pointer to a type that is extracting citation metadata on the fly. Currently, data providers need to maintain two versions of metadata which is not efficient.

## 2.9 Layered Service Components

Given that various information types as indicated above are being associated with the PIDs, a variety of useful layered services can be thought of such as:

* Find duplicates of digital objects in the domain of registered DOs and add information to the existing records if duplicates have been found. This would require to add information that can indicate identity.
* Find specific objects of interest based on specific attributes.
* Run processes that can find how many DOs created by a specific author have been cited or re-used by other researchers.
* etc.

## 2.10 PID Certification

Our dependency on PID systems and PID service providers is increasing, i.e. we need to have mechanisms to ensure the quality of services. It is urgent to extend DSA/WDS specifications by requirements which PID systems and service providers need to fulfil and execute checks at regular intervals. Some requirements have been defined under DFIG (<https://rd-alliance.org/group/data-fabric-ig/wiki/recommendations.html>) but they require broad agreement.

## 2.11 Workflow Initiation Component

This component carries out in principle similar tasks as described in 2.1 however it is being executed in workflows that combine and use existing data and generate new data, i.e. existing metadata etc. can be used to create new metadata. This component will use the existing metadata, provenance and PID information to create new metadata, provenance and PID records as indicated schematically in the diagram.



The details need to be worked out carefully, but it is obvious that providing such a component that supports the indicated flow of information is urgently needed to guarantee automatic documentation of processes. It seems that there are two ways to manage provenance records: 1) it is treated as a separate entity with its own attribute in the PID record or 2) it is treated as a subset of the metadata information. Since it is created at a different phase for different purposes we tend to prefer the first option, but perhaps both cases (and perhaps others) need to be provided.

1. Repositories are meant here in the abstract sense as a management instance having an API and internal procedures. [↑](#footnote-ref-1)
2. DOs can be individual objects, collections of DOs and metadata objects are also DOs. [↑](#footnote-ref-2)
3. fingerprint information to check identity and integrity, size of bitstream, type of object, etc. [↑](#footnote-ref-3)
4. PIDs of metadata record, landing page, etc. [↑](#footnote-ref-4)
5. Metadata records are being changed over time. Mechanisms must be in place to cope with these dynamics. [↑](#footnote-ref-5)
6. The term „system metadata“ is being used here although it is not clearly defined. [↑](#footnote-ref-6)
7. According to the DFT model „digital collections“ are digital objects. This is important in this context, since most operations will be carried out on collections. [↑](#footnote-ref-7)