Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 146 (2019) 123–130

Procedia Computer Science

www.elsevier.com/locate/procedia

14th International Conference on Current Research Information Systems, CRIS2018.

Decentralized Persistent Identifiers: a basic model for immutable handlers

Miguel-Angel Sicilia*, Elena García-Barriocanal, Salvador Sánchez-Alonso, Juan-Jose Cuadrado

Computer Science Department, University of Alcalá, Polytechnic building, Ctra. Barcelona km. 33.3, 28871 Alcalá de Henares (Madrid), Spain

Abstract

Persistent Identifier (PID) systems have evolved in the last decade to mitigate the problem of link rot and provide unique and resolvable identifiers for digital objects, becoming a key element in archival services. However, they still depend on centralized services which may either become unavailable or cease to behave or be managed as expected. Decentralized technologies may serve as the infrastructure for the design of a trustless PID system over which other incentive mechanisms may be devised. Here we discuss a basic model for such a novel PID system, based on the capacity of content addressing of current decentralized file systems and the concept and definition of verifiable claims as metadata statements about the existence of an immutable digital resource. An example basic design is described using the Interplanetary File System (IPFS) as supporting infrastructure with a minimal set of conventions, that could be used as an alternate mechanism for storage of byte streams in digital repositories as DSpace or CKAN.

Keywords:
Persistent Identifier Systems, decentralization, IPFS

1. Introduction

Different Persistent Identifier (PID) Systems have been developed to date as a solution for long-lasting reference to digital objects. These include for example the Handle System – subsuming the Digital Object Identifier (DOI) system –, the Persistent URL (PURL) or the Archival Resource Key (ARK) [8]. They are in widespread use nowadays and also commonly supported by digital archival systems and research information systems alike.

PID systems require an underlying infrastructure supporting the assignment, maintenance and resolution of the identifiers to the actual digital objects, as no identifier is inherently persistent. If the resolution system fails or stops working, the identifiers are no longer usable for applications, even if they are still accepted by their communities. Further, system security, reliability, and availability have been found to be perceived as significantly correlated to satisfaction with using PIDs [7]. A potential problem is that the services

*Corresponding author
Email address: msicilia@uah.es (Miguel-Angel Sicilia)

10.1016/j.procs.2019.01.087

behind these systems are centralized, either administered by a single organization or institution, or spread across many servers, but still relying on trust on particular providers. This raises different challenges for the long-term integrity, security and availability of PID systems. However, this may not be an issue if the sustainability of the supporting organizations is secured by a model of financing that reasonably guarantees their long-term operation.

In other direction, current PID systems rely on Internet services managed and hosted as any other service. In some cases, the databases of resources are hosted by a single organization, and in others, a central organization manages some form of namespace and assigns them to other organizations or individuals. For example, the persistent uniform resource locator (PURL) system, now hosted by the Internet Archive[1], allows a user to create a domain and then register PURLs as Web redirections, e.g. `http://purl.org/john` could be intended as a PID to a personal Web page. However, the actual availability of the resources is not supported beyond the responsibility and maintenance of that particular user, so that the mechanisms works simply as a way of creating a shared namespace that overlays on top of the Domain Name System (DNS) of the Internet.

Peer-to-peer (P2P) systems have been proposed as a solution in previous work [4] to the above mentioned problems. The main difference of these systems is that they are based on a network of computers that actively collaborate in a common goal, which in this case would be the maintenance of the digital objects beyond the capabilities of the individual user or organization. This way, long-term availability is shifted from the responsibility of a single organization or federated users to a network of collaborating computers, as it is done in cases as BitTorrent that result in a different approach to availability [11]. This in turn raises the need of an appropriate incentive system for participation [9], but in any case changes radically the scenario of curation responsibilities of digital objects that has to be reconsidered.

However, decentralizing PID systems requires also a means to identify provenance in a trustless environment (in which anybody, with or without identification is able to deposit resources), along with some conventions to describe the means of access to the entity referenced that allows for sorting them out and enabling their rendering and/or use via appropriate software tools. It is also important to consider the extent to which the decentralized approach supports good practices in identification [6].

Here we describe how emerging decentralized file system technologies as the Interplanetary File System[2] can support the design of a PID system when combined with simple and minimal conventions, that may be complemented with metadata oriented to interoperability (that represents an orthogonal concern [5]). That combination can also be used to deploy decentralized, highly generic solutions to metadata beyond identification and use, as described in [3].

The rest of this paper is structured as follows. Section 2 describes a model of persistent identifiers with the minimum commitments to make it universally applicable. Then, Section 3 discusses how that model could be implemented on top of a combination of a decentralized file system and a set of conventions. Finally, discussion and outlook is provided in Section 4.

2. Modeling Persistent Identifiers

This section discusses a basic model independent of any particular technology to clarify the underpinnings of the proposed kind of PID systems.

2.1. Base byte content model

The departure assumption for the model is that we have an infinite[3] space of **immutable** digital objects $o_i \in O$ that are distinguishable by their byte content[4]. A first layer for the system is that of being able to

[1] https://archive.org/services/purl/

[2] https://ipfs.io/

[3] In practical terms, this set is not infinite but arbitrarily large.

[4] It should be noted that this identification of objects with byte streams does not convey the usual understanding of intellectual works as conceptual creations, that are manifested in different realizations, but this will be discussed as a second layer later.

resolve objects from content-identifiers $i \in I$, and a function h defining a bijection $\forall o_a \in O; \exists i_b \in I, h(o_a) = i_b$ for which $h(o_a) = h(o_b) \rightarrow o_a = o_b$.

A basic practical requirement for an eventual deployment of this scheme is that given an object in O, we should be able to get its identifier (via h), and given one identifier, we should be able to retrieve the content object (we'll denote that process of retrieval with function h^{-1}). It should be noted that this is a mapping that affects to the byte contents, so that if many users ask for the identifier of different copies of the same object, they will get the same identifier. The implication is that depositing the same exact file (byte content) would be *idempotent*, i.e. repeated submissions of the same identical byte content will have no effect.

2.2. Layered metadata

Content identifiers do not carry any semantics, in the sense that the objects they refer to are not typed and opaque. This is problematic if actionable resources are required (e.g. those that can be rendered using common conventions and software packages according to their format). This requires a second layer providing metadata for resources.

That second layer can be modeled as a set of statements $\mathcal{S} \subset O$ that are associated to identifiers of **previously existing** objects by a function $p : \mathcal{S} \rightarrow I$. Note that this can be applied at several levels, as it is possible to have some statement associated to the identifier of a digital object that is in turn is metadata. This is essentially modeling a link from an object that represents metadata to an identifier of the described object. It should be noted that many objects in $\mathcal{S}$ may be describing the same digital object, forming Direct Acyclic Graphs (DAG) that have as "leaves" elements in $\mathcal{I}$ and "non-leaves" in $\mathcal{S}$. It is important to restrict the descriptions to avoid cycles, as otherwise we would end up having circular definitions. In addition to p, information needed for the retrieval and actionability of the referenced object should be embedded in statements, using some conventions. For simplicity, we considered that those sentences can be modeled as key-value pairs (k, v). This can be denoted as a mapping $m : \mathcal{S} \rightarrow \mathcal{P}((k, v))$. The domain of keys and values are intentionally left unspecified to allow for any form of metadata.

2.3. Metadata claims

The model so far allows for uniquely identifying objects at the byte level, and to associate metadata to them. If the identifiers of objects in $\mathcal{S}$ are (conventionally) used as pointers to resources, we have a first model of persistent identifiers. Those content identifiers are completely decentralized in the sense that act just as pointers to byte content but they can be arbitrarily and anonymously created with no restrictions.

An identifier system needs a model of trust (understood as the potential of verification of who stated what), as claims on identifiers (and in general on metadata descriptions) are made by agents (individuals or organizations), and as such require provenance. This should be kept as a separate concern, so that for a claim $c \in C$, where $C \in S$ will be the set of identities that are backing the claim.

Digital signatures provide the basis for attestation of a claim by an identity, and systems that connect those cryptographic identities with real world persons or organizations as digital certificates provide the link with liable agents when required.

If we restrict our discussion to persistent identifiers, a claim is an immutable record that can be modeled as a set of statements that include some particular ones in $m(c)$. Concretely:

- A timestamp.
- A digital signature from the entity depositing the claim.
- Some conventional statement that the resource is intended as a PID, e.g. a $(type,' PID')$ key-value pair.

Timestamps may be required for metadata in general, to account for conventions on the interpretation of repeated claims by the same agent, e.g. if there are two conflicting claims, which of them shall be regarded as the latest valid.

2.4. Shared conceptualizations

The above described model does not carry any inherent semantics. This is where the model calls for some kind of Knowledge Organization System (KOS) for coding types and a network system or relations that partially maps our social understanding of the documents.

This does not require a single, unified KOS, so more than one may coexist and even be (logically) inconsistent if representing, for example, different theories. This is also the place for standard or shared models. However, as PID systems have been traditionally developed for intellectual works, conceptual models as the Functional Requirements for Bibliographic Records (FRBR) [2] could be used as a shared, common convention.

Note that these systems may themselves be persisted and identified as objects in O, and then a basic content form of a metadata claim may be a set of tuples $(id_s, predicate, id_o, selector)$, where id_s may be a resource and id_o the identifier for a KOS, and the *selector* may be a reference local to the KOS (e.g. a particular concept or term). The Resource Description Framework (RDF)[5] is a kind of that model that brings the possibility of providing arbitrary descriptions[6]. However, this would require additional conventions, that can not be modeled with function p as described above, and we will not deal with them here.

3. Deploying persistent identifiers

The key implementation problem revolves around supporting functions (h, h^{-1} and p) preserving the properties required by an actionable resolution system, and a way for agents to deposit both objects in O and/or claims in C in a way that provides guarantees of provenance.

3.1. Content identifiers

Content identifiers can be realized by content hashing. This brings the important constraint as there can't be cycles in metadata relations, so the metadata layers are strictly a DAG, but this does not seem problematic as metadata is rarely refering to objects that in turn refer to the same metadata record. Distributed file systems as the Interplanetary File System (IPFS)[1] provide decentralized maintenance of digital objects retrievable by content hashing, thus implementing functions h and h^{-1} as built-in.

For example, if we add the current folder to IPFS with the command `add`, the file system will provide the content hash for each of the files and one for the folder as an aggregate (with sub-folders being assigned also hashes recursively if any). The following is an example output fragment for the command (content hashes are shortened for legibility).

```
>ipfs add -r .
added QmSTuTEThyESv...HPD8wSHVy7 test-ipfs/donut.jpeg
added QmSR9MJ5resQLj...jZeZDtW test-ipfs/purse.jpeg
added QmUNLLsPfHbsf67hvA3Nn test-ipfs/folder1
added QmaKZ3dnc9ejBdGg...Ljk6gXUnk9sdM9 test-ipfs
```

The links to other objects (the relation of a folder with its sub-folders or contained files) are embedded, forming a structure named Merkle DAG. In consequence, content hashing in IPFS fulfills the requirements for functions h, h^{-1}, the latter using some retrieval functionality as `get`. This solves the requirements and allows for composite objects to be retrieved and automatically have content identifiers. However, this by itself does not solve neither the requirements for provenance nor the permanent storage of resources that will be discussed below.

[5] https://www.w3.org/RDF/

[6] In RDF, it is also possible that the object of a predicate is an embedded literal and not a reference

3.2. Claims and links

The InterPlanetary Linked Data (IPLD) set of conventions allows for embedding IPFS hashes as links inside fragments of documents. In consequence, for example, an JSON file with embedded IPLD link implements function *p* and can be associated with arbitrary structured information providing support for function *m*.

The links may come with embedded information, which could give a first approach to a PID as sketched in the following JSON fragment that would be a representation of an element in $\mathcal{S}$:

```
{
   ...
  "pid": {
    ...
    "Content-Type": "application/pdf",
    "charset" : "utf-8",
    "doi": "10.1126/science.169.3946.635",
    "p": {"/": "/ipfs/QmUmg7BZC1Y..."}
  }
}
```

In the above example, metadata could follow any existing conventions, e.g. including elements in the DOI Data Model[7] or a combination, but it is advisable that the `pid` element surrounding the actual link `p` provides minimal information, not covering additional metadata that does not need to be associated directly with the PID, but could later be associated with the content hash representing that PID. Some of these elements describe how the resource is to be processed, in the example, this was done with standard HTTP header tags.

If the above fragment is added to IPFS, the content hash of that fragment constitutes the claim of existence of a digital object. That digital object is uniquely identified by its own content hash embedded in the `p` element, so that it can be dereferenced, but as the file system is P2P and decentralized, there is no trace of who deposited the metadata sentence.

3.3. Verifiable claims

Provenance in the Internet in a trustless infrastructure currently can only be provided by a combination of asymmetric cryptography with a tamper-proof immutable decentralized store. The link to physical world identities, if required, can be provided by associating claim signatures with digital certificates, introducing an upper layer of trust.

The W3C recommendation on Verifiable Claims Data Model and Representations[8] provides a way of expressing digital claims that can be tailored to the needs of the IPFS distributed file system as the supporting storage layer. A verifiable claim is a claim that is effectively tamper-proof and whose authorship can be cryptographically verified (e.g. contains a digital signature). Claims, according to the working draft specifications are statements made by an entity about a subject. This is a general purpose model, but we can constraint its meaning to adapt to our model of PIDs with the following:

- According to the draft, an entity is a thing with distinct and independent existence such as a person, organization, concept, or device. We could restrict this to organizations or persons, as PIDs are artifacts that are trustable to the extent the issuer is recognized by users as legitimate.
- Subjects are entities about which claims may be made. Here we restrict ourselves to digital objects, for which a PID is purposefully being added.

[7] https://www.doi.org/doi_handbook/4_Data_Model.html

[8] https://www.w3.org/TR/verifiable-claims-data-model/

In addition to that restricted interpretation of the verifiable claim, it is important to adopt a convention to tag PIDs. In our case, the `type` in the following example can be used to mark claims that are intended as PIDs. The following JSON fragment shows an example of a PID.

```
{
"@context": ["https://w3id.org/security/v1",
             "fs://ipfs/Qre...", ...],
"id": "https://bnf.fr/credentials/3732",
"type": ["Identifier", "pid-ipfs"],
"issuer": "https://bnf.fr", "issued": "2018-01-01",
"claim": {
    "id": "/ipfs/QmUmg7BZC1YP1...",
    "link": {"/": "/ipfs/QmUmg7BZC1YP1..."},
    "ark": "ark:/12148/bpt6k5834013m",
    "Content-Type": "application/pdf",
    "charset": "utf-8",
},
"signature": {...}
}
```

In the example, we have a claim with the following elements:

- The `@context` declaration is intended to reference definitions of terms, as specified in JSON-LD[9]. In this case, there is a reference to the security vocabulary[10] that defines `signature`, and a hypothetical reference to a vocabulary for IPFS PIDs stored in IPFS itself (hence the URL starting with `fs://`).
- The `id` is not the PID but another ID using conventional URI-based naming.
- References to the identifiers of the issuer and the issue date of the claim.
- The claim itself with information for rendering the digital object and the link itself to IPFS. In this case, the `id` of the claim is the same content hash of the link but it may alternatively be other. There is also an alternate hypothetical mapping to an ARK identifier.

It should be noted that it would be the content hash of the above JSON file that will constitute the PID, not the file itself or any other of the hashes. In this way, the retrieval of the claim would dereference the metadata associated, and a second operation of retrieval using the `link` attribute will be needed to access the digital object.

The infrastructure just sketched provides the basic facilities, but they should be complemented by retrieval or indexing engines that give the user either the reference to the claim in the blockchain or directly the content hash of the link. In the latter case, if the user or system requires some form of checking, it should revert to the former in which the signer of the transaction may be associated with a digital certificate.

3.4. On integrating with repositories

The PID model described could be integrated as a plugin solution in existing digital repositories. We will mention here some elements of CKAN as an example, as it provides well devised extension APIs, but the discussion could be applicable to similar repositories.

The typical workflow to register a dataset in CKAN asks for filling the metadata of the overall dataset, and then adding some resources (the actual data) that have also associated metadata. As metadata and the resources themselves can be changed or modified freely, the object is considered to be in a "draft" state, as depicted in Figure 1. In this state, the dataset is a valid CKAN dataset but has no decentralized PID and is not deposited in IPFS.

[9] https://json-ld.org/spec/latest/json-ld

[10] https://web-payments.org/vocabs/security

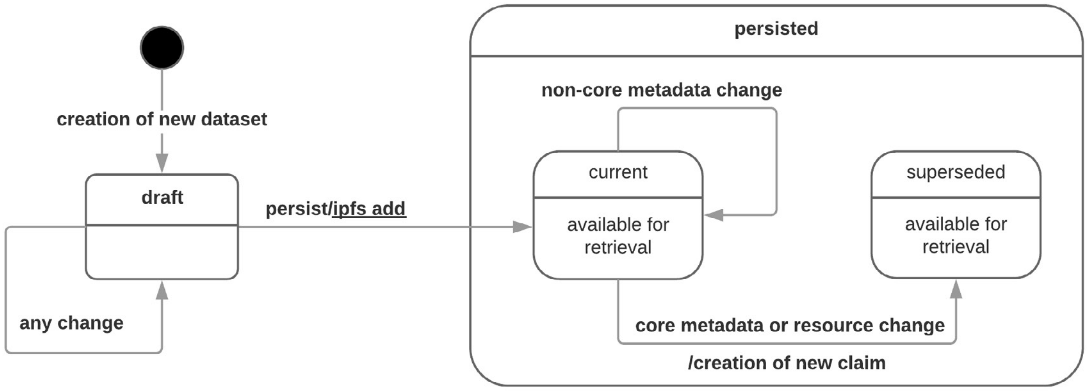

Fig. 1. State diagram of a verifiable claim as it moves to persistent ID in IPFS from CKAN

Once the metadata and resources are complete, an action of persisting the resource would entail interaction with the IPFS node and adding the resources and then the verifiable claim. The returned content hash for the claim, and the state of the claim can be stored as CKAN *custom fields*.

It should be noted that the removal or modification of resources, or resource metadata that was included in the claim itself (that we call here "core metadata") would modify the byte content of the object and/or verifiable claim, so a new PID should be generated. This could appear as a drawback at first glance, but it is actually an important underpinning of what a PID system requires. The previous PID would still be available (to the extent the P2P network does not "forget" it) so that it is still dereferenceable, and a link either at CKAN database or inside the claim of the new PID or both can be included to trace to the previous version.

The integration shall proceed with capturing the events on the modification of resources. Taking CKAN as example, this can be done by implementing the `DomainObjectModification` interface, that captures the events of update, deletion or modification of *Datasets* and its associated *Resources*.

This brief description of a possible integration reveals that moving to a decentralized storage and identification system may require changing the practice of archival description and update. These implications are still to be explored in future work.

4. Conclusions and outlook

Decentralizing the resolution of PIDs brings the benefit of decoupling PID systems from trusted parties or the responsibility of a federation of individually acting users, making them more robust, tamper-proof and opening the possibility of building decentralized incentive systems for making them sustainable in the long term.

A basic model and example deployment for PIDs not requiring reliance on centralized systems has been proposed. The system provides decentralized, immutable storage of objects and PIDs referring to them, together with an additional layer providing cryptography-based provenance for the attestations based on the idea of verifiable claims.

The same approach to depositing metadata for PIDs can be applied in general to archival functions that are the core of institutional repositories (IR), research information systems or their combinations [10], and potentially to other kinds of identifiers that are not referencing digital but physical entities.

The proposed model and the outline of a possible deployment should be considered just as a first incomplete attempt to devise a new kind of PID systems. Future work should proceed in different directions. On one hand, if we take the infrastructure and technical interoperability for granted, the main research direction is that of incentive systems and the extent to which different mechanisms support long term preservation and align with current archival practice and are acceptable by users. On the technical side, there is a need to propose and test deployment alternatives and conventions for interoperability. The integration of these systems with current digital repository software as DSpace or CKAN appears as a promising way of testing technical approaches to make these PID models work in practice. In particular, the archival management of digital resources in trustless environments also requires versioning and services for retrieval that would entail the use of more complex mechanisms. Previous work [3] has proposed the use of smart contracts over blockchains for such management functions, that can be integrated with models of PIDs as the one proposed in this paper.

References

[1] Benet, J. (2014). IPFS-content addressed, versioned, P2P file system. arXiv preprint arXiv:1407.3561.

[2] Carlyle, A. (2006). Understanding FRBR as a conceptual model: FRBR and the bibliographic universe. Library Resources & Technical Services, 50(4), 264.

[3] García-Barriocanal, E., Sánchez-Alonso, S. & Sicilia, M. A. (2017). Deploying Metadata on Blockchain Technologies. In Research Conference on Metadata and Semantics Research (pp. 38-49). Springer.

[4] Golodoniuc, P., Car, N.J. & Klump, J., (2017). Distributed Persistent Identifiers System Design. Data Science Journal. 16, p.34. DOI: http://doi.org/10.5334/dsj-2017-034

[5] Llanes-Padr/'on, D., & Pastor-Sánchez, J. A. (2017). Records in Contexts: the road of archives to semantic interoperability. Program, 51(4), 387-405.

[6] McMurry, J. A., Juty, N., Blomberg, N., Burdett, T. et al. (2017). Identifiers for the 21st century: How to design, provision, and reuse persistent identifiers to maximize utility and impact of life science data. PLoS biology, 15(6), e2001414.

[7] Park, S., Zo, H., Ciganek, A. P.& Lim, G. G. (2011). Examining success factors in the adoption of digital object identifier systems. Electronic commerce research and applications, 10(6), 626-636.

[8] Peyrard, S., Kunze, J. A. & Tramoni, J. P. (2014). The ARK identifier scheme: lessons learnt at the BnF and questions yet unanswered. In International Conference on Dublin Core and Metadata Applications (pp. 83-94).

[9] Rahman, R., Vinkó, T., Hales, D., Pouwelse, J. & Sips, H. (2011). Design space analysis for modeling incentives in distributed systems. ACM SIGCOMM Computer Communication Review, 41(4), 182-193.

[10] Rybinski, H., Skonieczny, L., Koperwas, J., Struk, W., Stepniak, J. & Kubrak, W. (2017). Integrating IR with CRISa novel researcher-centric approach. Program, 51(3), 298-321.

[11] Ye, L., Zhang, H. L., Zhang, W. Z. & Tan, J. (2009). Measurement and analysis of BitTorrent availability. In Parallel and Distributed Systems (ICPADS), 2009 15th International Conference on (pp. 787-792). IEEE.