

Decentralized Persistent Identifier Design: a basic model for immutable handlers (extended abstract)

Miguel-Angel Sicilia, Elena García-Barriocanal,
Salvador Sánchez-Alonso and Juan-José Cuadrado

March 4, 2018

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. However, they still depend on services that may become unavailable or cease to behave as expected. Decentralized technologies may serve as the infrastructure of a trustless PID system. Here we discuss their design and provide an example test deployment using a combination of IPFS and smart contract transactions over a public blockchain.

Keywords: Persistent Identifier Systems; decentralization; IPFS; Ethereum

1 Introduction

Different Persistent Identifier (PID) Systems have been developed to date as a solution for long-lasting reference to digital objects. These include Digital Object Identifier (DOI), the Handle System or Archival Resource Key (ARK) to name a few.

PID systems require an underlying infrastructure supporting the resolution of the identifiers to the actual 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. The problem is that the services 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 and availability of PID systems. Peer-to-peer (P2P) systems have been proposed as a solution in previous work [4]. However, decentralizing PID systems requires also a means to identify provenance in a trustless environment, 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.

Here we describe how emerging decentralized file systems and blockchain technologies can support such design when combined with simple and minimal conventions. 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 public blockchain. Finally, discussion and outlook is provided in Section 4.

2 Modeling Persistent Identifiers

2.1 Base byte content model

The departure assumption for the model is that we have an infinite¹ space of **immutable** digital objects $o_i \in O$ that are distinguishable by their byte content². A first layer for the system is that of being able to 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 this is that given a 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}).

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 $S \subset O$ that are associated to identifiers of **previously existing** objects by a function $p : S \rightarrow I$. 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.

In addition to p , information needed for the retrieval and actionability of the referenced object should be embedded in statements, using conventions modeled as key-value pairs (k, v) . This can be denoted as a mapping $m : S \rightarrow \mathcal{P}((k, v))$.

¹In practical terms, this set is not infinite but arbitrarily large.

²It 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.

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 associate metadata to them. If the objects in $\mathcal{S}$ are (conventionally) used as pointers to resources, we have a first model of persistent identifiers. However, those content identifiers are neutral in the sense that act just as pointers to byte content but they can be arbitrarily and anonymously created.

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 $c \in C$ there could be a set of identities that are backing the claim.

Those claims may not necessarily be in the same space as the identifiers of the objects, as they typically identify the agents (be it humans or machines) that deposited claims on the system by some kind of address (*addr* in what follows). 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 tuple $(id, timestamp, addr)$, where $id \in I \wedge \exists o \in O \ p(h^{-1}(id)) = h(o)$. In other words, the identifier points to an object that in turn references another one. 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, but several of them may co-exist 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)³ is a kind of that model that brings the possibility of providing arbitrary descriptions⁴. However, this would require additional conventions, that can't 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 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.

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 Directed Acyclic Graph (DAG), but this does not seem problematic as metadata is rarely referring 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.

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 m .

The links may come with embedded information, as in the following example:

```
{
  ...
  "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 Model⁵ 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. Some of these elements describe how the resource

³<https://www.w3.org/RDF/>

⁴In RDF, it is also possible that the object of a predicate is an embedded literal and not a reference

⁵https://www.doi.org/doi_handbook/4_Data_Model.html

is to be processed, in the example, this was done with standard HTTP header tags.

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. Current Turing-complete blockchains as Ethereum allow for the implementation of such infrastructure, and the link to physical world identities, if required, can be provided (when required or desired) by associating blockchain signatures with digital certificates, introducing an upper layer of trust.

At a basic level, a smart contract on Ethereum allowing for recording simple claims as `is-pid` is sufficient. The claim should simply be a record of the transaction sender (which may be an author, a publisher or other stakeholder in archiving), a timestamp (if the block number is not considered enough) and the link to the statement hash residing at IPFS. That would require the use of an oracle with current Ethereum technology.

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 attestation checking, it should revert to the former in which the signer of the transaction may be associated with a digital certificate.

4 Conclusions

Decentralizing the resolution of PIDs brings the benefit of decoupling PID systems from trusted parties, 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.

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>