DATA SCIENCE JOURNAL

Car, N J et al 2017 The Challenge of Ensuring Persistency of Identifier Systems in the World of Ever-Changing Technology. *Data Science Journal,* 16: 13, pp. 1–18, DOI: https://doi.org/10.5334/dsj-2017-013

RESEARCH PAPER

The Challenge of Ensuring Persistency of Identifier Systems in the World of Ever-Changing Technology

Nicholas J. Car[1], Pavel Golodoniuc[2] and Jens Klump[2]

[1] Geoscience Australia, Canberra, ACT, AU

[2] CSIRO Mineral Resources, Perth, WA, AU

Corresponding author: Nicholas N. J. Car (nicholas.car@ga.gov.au)

The identification of information objects has always been important with library collections with indexes having been created in the most ancient times. Since the digital age, many specialised and generic persistent identifier (PID) systems have been used to identify digital objects. Just as many ancient indexes have died over time, so too PID systems have had a lifecycle from inception to active phase to paralysis, and eventually a fall into oblivion. Where the indexes within the Great Library at Alexandria finally succumbed to fire, technology change has been the destroyer of more recent digital indexes.

We distil four PID system design principles from observations over the years that we think should be implemented by PID system architects to ensure that their systems survive change. The principles: describe how to ensure identifiers' system and organisation independence; codify the delivery of essential PID system functions; mandate a separation of PID functions from data delivery mechanisms; and require generation of policies detailing how change is handled.

In addition to suggesting specific items for each principle, we propose that a platform-independent model (PIM) be established for persistent identifiers – of any sort and with any resolver technology – in order to enable transition between present and future systems and the preservation of the identifiers' functioning. We detail our PID system—the PID Service—that implements the proposed principles and a data model to some extent and we describe an implementation case study of an organisation's implementation of PID systems that implement the Pillars further but still not completely.

Penultimately, we describe in a Future Work section, an opportunity for the use of both the Pillars and the PIM; that of the World Wide Web Consortium's Permanent Identifier Community Group who is seeking to "set up and maintain a secure permanent, URL re-direction service for the web".

Keywords: persistent identifier; governance; URI; Handle; DOI; design principles; platform independent model

Introduction

The volume, range of types and sophistication of information (data items) collected and managed by society are growing faster than at any point in history,[1] however the persistence of that information may be decreasing (Kuny 1998). While the traditional means for storing information, such as in books,

[1] Buringh and Van Zanden (2009) chart the great increases in printed manuscripts and books from the 6th to the 18th centuries. Larsen and von Ins (2010) chart the rate of growth of scientific publications in the 20th century and numerous sources chart the exponential growth in digital data in the early decades of the 21st century, such as King (2011) and Stadd (2013). The studies, taken as a whole show both the huge growth in information volume (both total and per person) and also the growth in information type. King (2011) shows the growth in Short Messaging Service messages, and Twitter "tweets" and Stadd (2013) charts growth in websites and videos as well, thus the growth in type range of information too. Some of the new information forms, websites, tweet conversations etc. are also more sophisticated than manuscripts in terms of their multiple authors, multi-message interrelations and potential text and non-text content (images and video) as well.

on parchment scrolls or even clay tablets are still available to us, the rise of digital storage is clear: digital is how the world mostly does, and increasingly will, store its information. For information to persist usefully, whether in traditional or digital forms, it needs to at least be identifiable, accessible and readable.

Clay tablets were collected, identified and indexed at least as far back as 630BC[2] but likely much earlier than that, back to perhaps 2,000 or 3,000 BC (Finkel & Taylor 2015) and identifier mechanisms for books, such as the International Standard Book Number (ISBN) (R.R. Bowker LLC 2014), have been in place for almost half a century to assist with book identification and, via library indexes, accessibility.

For digital data, many identifier mechanisms have also been implemented. Some of these not only identify data items, as ISBNs do, but also provide resolution mechanisms to give users direct access to digital copies of the data identified or, at least, to systems that can give them access. This is possible due to identifier resolver networks and the fact that, unlike books and scrolls, copies of digital data can be delivered remotely via computer networks including the Internet.

While identifiers for digital data items clearly have many advantages over those for physical data items, they also have some negatives. The persistence of indices of digital identifiers is itself digital and requires computer infrastructure to operate which must be paid for and managed.

They are therefore more fragile, or likely to be more fragile, than, say, a library's simple card-based Dewey Decimal (Dewey 1876) book index. Web page addresses – a commonly used form of digital data item identifier – often have short lifespans even when the information they link to is persistent. Zittrain *et al.* (2013) found that, in 2013, of 555 web page URL links cited in USA Supreme Court opinions delivered in the preceding 7 years, 36.4% did not resolve at all and 49.9%, while resolving, no longer returned the cited material.

Trustworthiness of Persistent Identifier Systems

In the past couple of decades, a number of persistent identifier systems have been developed. These PID systems were developed by various communities and had overlapping aims. Over time, they proved to be successful to various degrees and even expand into new applications due to their generic nature and inherent flexibility. Nevertheless, all the systems exhibited one common trait – they seemed to follow the same lifecycle from their inception and active phase through to paralysis and eventually the "zombie stage" (Beck *et al.* 2016; Huber & Klump 2016) in which they continue to exist but contain no value.

Two of the most notable examples of once solid and trustworthy systems that are slowly but surely sliding into the oblivion are the Life Science Identifier (LSID) system initiated by the Object Management Group (OMG 2004) and the Persistent URL (PURL) (Internet Archive 2016) originally developed and commissioned by the Online Computer Library Center (OCLC) in 1995. The LSID lacked a robust identifier resolution system that, to some extent, impeded its uptake by the broader community. The system gradually faded away leaving a legacy of 5M+ unresolvable identifiers.

The PURL system, although initiated by the librarian community for their use, proved to be generically applicable and simple enough to find its way into multiple science disciplines making it truly domain-agnostic. The possibility to instantiate a private instance of the PURL server also contributed to its success. Like DOIs, PURLs do not offer content negotiation or query string handling meaning that they support none of the advanced features of HTTP URIs that allowing a single PID to dereference to different forms of a resource. The main OCLC-managed PURL system at purl.org which was by far the most used installation lacked community financial support due to a missing financial model, unlike DOI. With its pros and cons, PURL proved to be very low-maintenance and cheap (free) and ultimately successful solution for many and generated over 100,000 identifiers during its lifetime. Nonetheless, the OCLC eventually lost interest in active support of the system and PURL slid into the paralysis phase as of November 2015, with eventual rescue by the Internet Archive in 2016 (OCLC 2016).

What once seemed to be an obvious candidate for persistent identifiers, HTTP URIs implementing the 'Cool URI' principles (Berners-Lee 1998), given the intentions of those principles to promote persistent, system-independent identifiers, seems less obvious when one considers that systems and agreements on

[2] Ashurbanipal created a library of cuneiform tablets in ancient Ninevah collected from all over his kingdom, some of which were several thousand years old at that point. He created shelf indexes for the tablets too. See the Encyclopedia Britannica article on Ashurbanipal by Wiseman, a noted Assyriologist, for more information on the ancient collector. https://www.britannica.com/biography/Ashurbanipal

which they are dependent, such as the Domain Name System (DNS) (Mockapetris 1987)[3] might vanish and be superseded by new technologies. Already, the governance of DNS has changed (Ribeiro 2016), so we may confidently say change is inevitable, despite the best endeavours of the Cool URI designers to avoid sources of change.

Bütikofer (2009) devised a set of guidelines and criteria to assess the 'trustworthiness' (i.e. likelihood of persistence) of digital persistent identifier systems. The aim was to 'help providers and users of persistent identifiers keep digital objects identifiable, referenceable and accessible over longer periods and despite unforeseeable changes'. In order to create identifiers that are not reliant on a single system, as HTTP URIs are with global DNS, PID systems with decentralised resolution infrastructure have been made. This allows PID resolver implementers to use any technology they wish: they need only adhere to a protocol. Notable examples of distributed resolver systems are Digital Object Identifiers (DOI) (International DOI Foundation 2016), Handle (CNRI 2016) and Magnet Links (Mohr 2002)[4] that use Internet-based Distributed Hash Tables (DHT) or Peer Exchange networks (PEX). The Internet itself is free from dependence on a single system given that it too is a decentralised network of nodes that adhere to a protocol, the Hypertext Transfer Protocol (HTTP) (Fielding *et al.* 1999). The Internet as a whole is not dependent on DNS but the commonly used web addresses (URIs and URLs) are. Systems such as DOI can thus support resolution mechanisms that are likely to be able to maintain the resolution of identifiers regardless of changes in technology or to one particular system.

As of today, with enough historical data, the observed negative trend seems to be repeatable and this caused many researchers (Beck *et al.* 2016; Golodoniuc *et al.* 2016) to rethink the whole approach to persistent identifiers and look into the ways to break that vicious cycle to answer a fundamental question of what makes a persistent identifier system trustworthy?

We propose to explore consequences of the potential loss of the foundation of most of existing PID systems and identify traits a PID system should exhibit to be as technology-independent as possible and be flexible enough so that identifiers and metadata associated with them can be migrated to future systems when needed.

PID Pillars

We propose a layer of abstraction in PID system design that separates persistent identifier systems from specific technical solutions, domain-specific requirements and even resolver system protocols. In this abstraction, we make recommendations for the establishment of four 'pillars' upon which, we believe, successful PID systems could be, or even already are, built. This work extends on the analyses of multiple identifiers schemes and the establishment of 'criteria for assessing the trustworthiness' of them (Bütikofer 2009).

Our four pillars, despite all being intrinsically interconnected and thus all necessary for successful PID systems can, we believe, be considered independently. The pillars are:

1. Identifier Independence
2. Delivering Essential PID Functions
3. Separation from Data Delivery
4. Employing Policies for Change

Identifier independence

Creating identifiers that are independent of any particular technology or organisation and are able to be unambiguously understood are well-known requirement for PID systems. The posit 'Cool URIs' was put forward for system and organisation-independent HTTP Universal Resource Indicators (HTTP URIs) (Berners-Lee 1998) and the Handle Technical Manual (CNRI, 2015, see the 'Handle Syntax' section) which articulates requirements for readability sating that identifiers must be:

- Any printable characters from the Universal Character Set of ISO/IEC 10646 (ISO 2012):
  - UTF-8 encoding is required;

[3] The Domain Name System is actually a result of numerous Request for Comment documents by the Internet Engineering Taskforce over time, starting with RFC 1034 (Mockapetris, 1987). See the 'updated by' section of the HTML version of RFC 1034 (https://tools.ietf.org/html/rfc1034) for links to the more recent RFCs.

[4] While a reference is provided to one of the Magnet URI scheme original documents, the Wikipedia article on the Magnet URI Scheme, https://en.wikipedia.org/wiki/Magnet_URI_scheme, is recommended as a better entry point to understand the scheme.

- Case insensitive:
    - Only ASCII case folding is allowed.

We extend the HTTP and Handle identifier requirements to 1) broaden them beyond a single protocol, and 2) to allow for pattern matching identifiers. The reason for broadening them is that identifier resolution systems may be forced to change protocols over time and what is acceptable for one protocol may not be for another. LSIDs (OMG, 2004), which are based on URNs, are resolved via HTTP URI resolvers today, for example, the LSID Resolution Project (International Working Group on Taxonomic Databases, 2017) and the Atlas of Living Australia (ALA, 2017). In the future, they may need to use other protocols (Mendelsohn, 2006), however the choice of character sets in the identifier could prevent the use of some protocols.

The reason for the implementing pattern matches is that while one-to-one (1:1) identifier matching has been the norm for URIs, Handles, URNs etc., implementation systems may depend on pattern matching to both interpret *identifiers*, and to implement identifier part replacement to re-write them to *locators*, as is the case for the commonly used Apache web server's rewrite module (Apache, 2017) for HTTP URIs. An example of part replacement could be a pattern such as http://{BASE_URI}/document/{DOC_PID} where the first 3 characters of the ID are a targeted delivery system's identifier and the remainder an individual document identifier. A length-specified replacement pattern may split the DOC_PID to allow the PID system to forward the individual document identifier to the target system. In doing this, the PID system may replace the individual document identifier with a full path relevant to the target system which, when combined with the target system's base identifier makes a complete, resolving URL.

The set of existing individual identifiers conforming to a pattern is not known before attempted resolution, only the complete set of all possible pattern matches. An example would be the pattern 0..100 which would match any one of one hundred and one integers from zero to one hundred. A system resolving that pattern may handle requests that match the pattern but do not result in a valid response with some sort of response code, much like the HTTP 404 Not Found status code (Fielding *et al.* 1999).

Given that various fields within a matching pattern may be used to construct the resource locator according to a replacement pattern, as per the example above, and that the full set of valid identifiers may not be known, a pattern-based identifier cannot be replaced with a set of 1:1 PIDs unless every possible pattern match is tested, including any replacement patterns, and existing individual PIDs recorded. While this is entirely possible and perhaps sometimes necessary if 1:1 matches only need to be supported, it is likely it could usually be avoided as long as the patterns are precisely defined, allowing for exact replacement.

Thus, our recommendations for identifiers are that they:

a. Avoid organisation names;
b. Avoid technology references;
c. Avoid resolution protocol indicators and characters problematic for well-known protocols;
d. Avoid visual ambiguity and use a well-known character set;
e. Define which, if any, pattern matching system use.

The first two points follow recommendations made for "Cool URIs" and recognise the face that organisation names change and technology evolves and becomes superseded, which leave an unwanted legacy of non-current identifier parts or, worse, unresolvable "zombie identifiers" (Beck *et al.* 2016; Huber & Klump 2016). It may not always be possible to maintain organisation-specific or technology-specific identifier parts, even if it is acceptable that they remain in use after name changes. For instance, some domain name ownership regimes require them to somewhat reflect organisations' business names which they may not do after an organisation name change.

Protocol independence is akin to an identifier being, perhaps, abc.def.ghi as opposed to http://abc.def.ghi, the latter of which indicates the use of the HTTP protocol. Characters well-known to cause issues in some protocols such as whitespaces, question marks, colons, backslashes etc. should also be avoided and, in the safest case, perhaps only letters and numbers used. We acknowledge that this is both a list that is unknowable in its entirety (which protocols does one cater for) and quite restrictive, however, the characters remaining for use include letters (both ASCII and even UTF-8) and numbers, the combinations of which are effectively infinite thus no functional restrictions on identifiers are made by this recommendation. By implementing this recommendation, identifier system owners may be able to bypass single system resolvers such as the Domain Name System (DNS) used by HTTP URIs and is prone to single points of failure, as recent DNS Denial of Service attacks have shown (Newman, 2016) and perhaps use new, even more distributed, resolver systems, such as Magnet Links (Farrell *et al.* 2013) as proposed in Golodoniuc *et al.* (2017) (this volume).

The avoidance of visual unambiguity will likely impose restrictions on the way identifiers are formed for human readability and may be of concern when direct human interaction with identifiers is essential, e.g., writing identifiers on physical materials, reading them for others to hear, etc. While character ambiguity is able to be avoided in computing environments through character encoding declarations, for example declaring that an HTML page uses UTF-8 characters by including the declaration `<meta charset=”utf-8”/>` in the document's header, it is unlikely that all users of identifiers will be able to know the encoding used. Some users may be isolated from any metadata associated with that identifier, perhaps before resolving it or reading a printed identifier off a physical sample's label, and, even if they can access metadata, they may not be able to visually differentiate characters within an encoding, such as different forms of whitespace, zeros and O's, minus signs, dashes and hyphens and similar Latin and non-Latin characters.[5] To ensure accurate human readability, specific communities of practice might impose rules on characters and character sets acceptable for identifiers.

When a pattern-matching identifier is used, the pattern matching system must be indicated. This is so any re-implementers of the system are aware of the exact matching syntax. Apache's Rewrite module uses patterns based on the Perl Compatible Regular Expressions (PCRE) (Hazel 2012) however other regular expression systems are used and expression languages change over time.[6] Additionally, when pattern matching is used for identifiers, it is common to subset the identifier for variable handling and such handling is less standardised than Regular Expression pattern matching syntax.

Delivering essential PID functions

Regardless of a specific design a PID system is ultimately responsible for management and handling aspects of persistent identifiers and ensuring the overall integrity of the system. The main tasks of any PID system are:

1. Issuing identifiers;
2. Storing identifiers;
3. Resolving identifiers.

How these are implemented will vary tremendously depending on design and technology choices, but these functions, essential to all PID systems, remain the same and must be implemented by any PID service claiming to be such.

We can posit some basic constraints on these three functions.

Issuing identifiers

Identifiers issued by PID systems need to adhere to the principles stipulated by the 'Identifier Independence' pillar, but a PID system also needs to ensure:

- **Uniqueness** – within some scope, not necessarily globally, to avoid clashes;
- **Ownership** – identifiers created must be able to have their management restricted to particular agent;
- **Editable metadata** – identifiers' metadata must be able to be edited in order to allow their owners to update details of the thing they are referring to, such as its location, as they will inevitably change.

Storing identifiers

Identifiers, once issued, must be stored by PID systems and maintained for a long time. Of course different lengths of time may be appropriate for different identifiers but it is reasonable to assume people make PIDs in order to persist for some sort notion of long compared with other timelines in the IT and electronic data worlds. This does not need to be restricted to a single storage system and may be distributed across a network of nodes where there is a continual turnover of some proportion of the nodes. The size of the PID datasets (consisting of identifiers themselves and their metadata) for the largest existing PID systems is not large in

[5] Security exploits have been made using similar-looking non-ASCII characters and these sorts of exploits are now referred to as an 'IDN homograph attack'. See https://en.wikipedia.org/wiki/IDN_homograph_attack.

[6] Note the discussion of changes in Regular Expression systems over about 20 years described by Barnett (2011).

comparison to some digital datasets, but is certainly non-trivial for some systems (the DOI system claims over 100 million identifiers registered (International DOI Foundation, 2016)). The size is also likely to grow greatly as persistent identifiers are starting to be used at finer resolutions than before; not just whole digital datasets but for things like granular Linked Data objects. Atkinson & Box (2016) and Chief Technology Officer Council (2011) provide examples of systems and directives respectively for making Linked Data URIs for many things.

Identifiers and their metadata should be stored in such a way that they can be, or automatically are, insulated from accidental data loss and from the possibility of not being able to be interpreted in the future. They should also have changes to their metadata recorded to cater for provenance.

From these two points above, we posit that PID systems should store identifiers while catering for:

- **Scalability** – in the data/metadata stored;
- **Integrity** – ensuring data is backed up or adequately replicated across nodes in a distributed system;
- **Interpretability** – ensuring data is understandable by adequately documenting it;
- **Versioning** – ensure that PID metadata can be retrieved as it was presented at certain times and that the provenance of identifiers can be discovered;

The lower orders of identifier interpretability require that data formats and encodings are interpretable, but higher order interoperability requires that creators and users of identifiers can understand what is and isn't stored by PID systems for identifiers. This is best handled by a PID metadata model and we propose one in the following Section.

Resolving identifiers

Identifiers must, at least, be able to be used to return the metadata stored with them. They might also, in the Internet world, be able to resolve or indicate the location(s) of the data of the thing that they are identifying and not just its metadata. This second goal is merely a subset of the first, when data locations and perhaps instructions as to how to interact with resources at those locations, are to be given only and not the actual data itself; locations and instructions can be stored as metadata. This distinction is further described next.

Separation from data delivery

While resolving identifiers to metadata is a core PID system task, delivering data for the resource identified is not. Unlike Bütikofer (2009), we believe making recommendations regarding the delivery of the information objects is outside the scope of a PID system. This is due to data services needing to provide their own interfaces and protocols for data delivery, which are likely to be domain-specific, possibly complex and possibly required to deal with large data volumes, all of which are able to be removed from the PID system's goals. We work with the assumption that, given persistent identifiers that are able to be resolved to metadata, there will always be a multitude of mechanisms available to access the data of the identified item, any one of which that metadata could indicate. Examples are direct file download links, calls to Application Programming Interface (API) functions, intermediate landing pages providing further links for data access, even person/organisation contact details to allow non-automated requests for data. The data mechanism can, and likely will, change over time, but the identifier should not.

This principle is not always employed by systems that both store data or provision space for it and mint identifiers: such systems can lock identifiers to a specific resolution mechanism for the item identified. One method of overcoming this is simply for systems to present a "landing page" at an item's identifier's resolution point which then may indicate multiple data access possibilities. An example of a system doing this is two of these authors' own organisation's "Data Access Portal" (CSIRO 2017) which is both a register and a repository. For datasets stored, such as "PROMS Server source code", in addition to direct access to the data (in this case, a zip file) a landing page containing metadata is created and addressed with a native identifier; the URI https://data.csiro.au/dap/landingpage?pid=csiro:13686. Additionally, a DOI is minted (in this case, 10.4225/08/5571046ED9FDB) which resolves to the system's landing web page and provides a further separation between the identifier and data resolution and delivery.

It is not necessary for registry/repository systems to separate their registered item data delivery only when one-to-one matches between a native identifier and an external identifier can be made, such as the URI and DOI situation described above. Pattern matching can be used, for example, a URI-based catalogue with identifiers such as http://catalogue.organisation.org/technology_detailsome/deep/folder/location/resolver.php?id={ID} may be resolved to by a persistent URI system with the URI such as http://persistent-uri.org/item/{ID}.

Employing policies for change

PID system creators should establish policies that define how the essential functions of the service can continue despite technology and social change until the point at which the system is deliberately ceased. Policies need to explicitly spell out how to deal with situations beyond control of the PID system itself that affect its operation. These include:

- **Technology change** – when technology it relies on must be changed for any reason;
- **Social change** – when key players in the PID system change or end their involvement;
- **Identifier abandonment** – when identifier owners stop maintaining the identifier's metadata;
- **Financial sustainability** – how the system will be financed for its expected lifetime;
- **Decommissioning** – when the total PID system needs to be decommissioned completely (not migrated).

These policy areas area similar in some respects to "Governance" and "Sustainability" principles espoused for scholarly infrastructures by Bilter *et al.* (2015). Directly relevant is their call for scholarly infrastructure owners to publish a "Living Will" which is a plan to handle system decommissioning and to have a wide range of stakeholders in their governance structure. Their detailed discussion of financial planning to ensure that private scholarly infrastructures have the incentives for dealing with technology change and data preservation motivated the "Financial sustainability" point above and further discussion of this important aspect of system planning below.

A general computer design principle that can be followed to assist with the first point above is that platform-independent data and function models for the PID system need to be required before production operation is allowed. These models then allow alternate technologies to be employed to match data and function when technology change is required.

A corollary to a model-based approach is to require the PID system to implement backup and restore or export and import functionality using a serialisation of the data model. The Persistent ID Service (Golodoniuc 2015a; Golodoniuc *et al.* 2015b) allows its entire database of identifiers and their editing history to be exported and imported using an XML serialisation of the creator's, a snippet of which is given in **Figure 3**.

A governance mechanism needs to be agreed on to ensure that the ceasing involvement of no one individual or organisation can break the system. The PURL system nearly ceased in 2016 when the original maintainer, the Online Computer Library Center, ceased to support the system (OCLC 2016).

When individual owners of identifiers cease to care about them, a process needs to be entered into in order to gracefully retire them, even without express permission of the (now uncaring) owner. It is important for dormant persistent identifiers to perform some resolution function as straightforward brokenness will damage trust in the system. While the HTTP protocol specifies a series of response codes (Fielding *et al.* 1999) that can be issued by web servers to indicate different sorts of resource (un)availability, none express a sentiment such as 'this resource used to exist but no longer does'. Thus, a PID system which just responds with an HTTP-style status (perhaps, 404 Not Found) will not suffice. Bütikofer (2009) suggests a 'qualified response which differs from a technical error message' for such cases and this is what is implemented for 'tombstoned' URIs by the Persistent ID Service, but that system does not give criteria for when to tombstone a URI. When and who implements tombstoning is thus the focus of this policy.

The financial sustainability of any system is an important factor in determining its longevity and how it can/cannot afford to handle change. Klump *et al.* (2015) discusses the initial and revised costs of DOIs and how, on the one hand, a costed model certainly pays for the system, it can lead to reduced adoption. A free to participate model is in place for other persistent identifier systems, such as the newly revived PURL (Internet Archive 2016) and IGSN (IGSN e.V. 2016) although the latter has community requirements for participants to contribute resourcing. Different funding models may be appropriate for different PID systems but a clearly articulated funding policy should be articulated at system inception.

A PID system should indicate technically and social how it should be decommissioned, when the need arises. This was absent from PURL's setup but has been recognised for other computer infrastructures such as by Bilter *et al.* (2015). While aspects of the other Pillars may help with decommissioning – for example data export – a policy must actually be formulated to trigger this. The W3ID persistent identifier discussion lists[7] have discussed

[7] See email discussion within the 'W3ID' mailing list at https://lists.w3.org/Archives/Public/public-perma-id/, particularly the 'Re: Problems and Opportunities at purl.org' thread in 2016 (e.g., https://lists.w3.org/Archives/Public/public-perma-id/2016Feb/0009.html) in which participants attempt to propose a platform-independent HTTP URI data model in a well-known format such as Comma Separated Values.

technological independence of potential future PID systems at great depth but have not suggested related policy or principles. The risk here is that an event that forces a system decommissioning (technology or social change) may find system participants and people relying on it with no clear roles or responsibilities and thus, despite technical opportunity, paralysis may ensue.

Implementing the Pillars

Many parts of the pillars described above are implemented by various identifier systems. The author's own Persistent ID Service (PID Service) (Golodoniuc 2015a; Golodoniuc *et al.* 2015b) implements some of the technical recommendations but not all (see **Table 1**); it was developed before these pillars were crystallised. The PID Service caters for identifier uniqueness, ownership and editable metadata and, due to its storage mechanisms, it implements good scalability, integrity, versioning and some level of interpretability through its use of a data model and API for its HTTP URI-based persistent identifiers. The PID Service somewhat separates itself from data delivery by allowing content negotiation and query string parameters to be used with its HTTP URIs in order to access the data of the identifier's target resource in any number of ways however this is limited to access of the resource's data via the HTTP protocol.[8] The PID Service does not, in and of itself, address social change, owner abandonment and other issues that require governance policies for their handling.

Pillar	PID Service Implementation
Identifier Independence	Cannot enforce syntax policy so *Avoid organisation names* – not enforced, *Avoid technology references* – not enforced. Locked to a single protocol (HTTP URIs) so unable to *Avoid resolution protocol indicators.* Uses a User Interface that will remove *characters problematic for well-known protocols.* Does *Avoid visual ambiguity and use a well-known character set* – implements UTF-8. Does *Define which, if any, pattern matching system they use* – the regex system in use is documented.
Delivering Essential PID Functions	Does *Issue identifiers* – part of the tool's User Interface and API. *Uniqueness* – enforces by relationship (hierarchy) and resolution checking. *Ownership* – recorded for every PID. *Editable* – PID metadata editable via UI & API. Does *Store identifiers* – the tool uses a database. *Scalability* – potentially limitless (given the use of a robust, scalable database). *Integrity* – a duty for the implementer. *Interpretability* – documented by the data model and system documentation. *Versioning* – automatically captured and stored. Does *Resolve identifiers* – if installed as recommended with web server functionality providing access.
Separation from Data Delivery	Inherent: no ability for the system to deliver data.
Employing policies for change	Mostly a task for the implementers however: *Technology change* – can be decoupled from a specific database, is loosely coupled from a front-end web server, thus certain components may easily be changed. *Social change* – unable to be addressed by a system. *Identifier abandonment* – identity of each identifier's owner stored. System admin has access to all. *Financial sustainability* – not addressed. The project was originally funded for development and some early adoption but no general funding for community development or use is yet arranged. Individual institutions have funded instances of the system in place. *Decommissioning* – documented and comprehensive export formats (the XML shown in **Figure 3**) assist with this.

Table 1: The extent to which the PID Service implements the four Pillars.

[8] This is more flexible than 'Any customer can have a car painted any color that he wants so long as it is black' (attrib. Henry Ford, c. 1909) since the HTTP protocol does allow for a range of data access mechanisms and can also, on many operating systems, be used to trigger data access via a range of well-supported protocols such as FTP but it is nevertheless limited to fewer mechanisms than the totality of what might be considered useful for network data access.

In the remainder of this section, we discuss implementing a more generalised data model for persistent identifiers' metadata going beyond the PID Service's HTTP URIs and implementing policies for change.

PID PIM motivation

The formulation of a platform-independent model (PIM) of PID metadata has been the focus of much recent discussion within some of the PID research community identifying itself as the 'W3ID' group. While there is a general recognition that PIMs are important for technology independent systems, the immediate motivation for the W3ID group's recent discussion was the cessation of support for the well-known (in sematic web communities) and long-lived PURL system's infrastructure by its sponsor, the Online Computer Library Center (OCLC) and the need to maintain the PURLs. The purl.org service has subsequently been maintained by the Internet Archive who have implemented new management systems for it allowing continued identifier resolution and management (Graham 2016).

Two of the PID PIMs suggested by the W3ID group are shown in **Figure 1**. These models contain some of the necessary information required for the HTTP redirection of one URI to another. This is the PID action model used by systems such as PURL where a domain name, purl.org in the case of PURL, is used to create URIs that, due to the management of the domain, are guaranteed to, or at least hoped to, not change. With the information available in the model given **Figure 1b** and implemented in **Table 2**, an HTTP URI redirect may be made for a single URI match. The use of patterns in place of fixed paths in **Figure 1b** and **Table 2** allow for pattern-based redirection. The PID information represented in **Figure 1** and **Table 2** is a great simplification of the information required to generate proxy server configuration for URI redirection, for example Apache server's Rewrite module.

PID PIM scope

Some identifier schemes require certain metadata to be associated with identifiers that is pertinent to their domain and not the functioning of the identifier. For example, DataCite-created DOIs, coming from the world of publication, require Identifier; Creator; Title; Publisher; and PublicationYear (DMWG 2014).

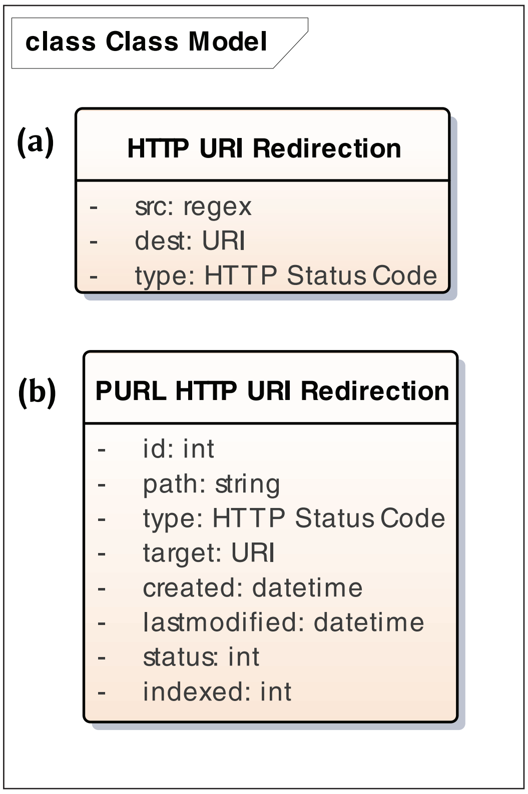


Figure 1: Two simple classes for HTTP URI redirection objects: a) simple URI redirection properties of a source pattern (src), a destination URI (dest) and an HTTP Status Code (type) (Soiland-Reyes 2016a); b) URI redirection for PURLs with some ownership metadata. 'URI' objects are valid HTTP URIs according to RFC2616 (Fielding *et al.* 2016) and 'Status Code' objects are valid Status Codes according to the same specification.

The scope of the PID PIM is only functionality and therefore domain-specific metadata elements are excluded.

Since these authors believe it is necessary to retain information about the Custodian of the PID in order to handle situations such as identifier abandonment, information such as Custodian is relevant to a PID PIM whereas Creator or Publisher is not. To ensure that information relevant to the functioning of the PID only is mandated, the agents related to a PID in the PIM named according to the standardised codelists for the description of metadata datasets compliant with ISO/TC 211 19115:2003 and 19139 (ISO, 2008).

The Persistent ID Service model

A more detailed HTTP URI PIM is implemented by the PID Service (Golodoniuc 2015a; Golodoniuc *et al.* 2015b). A UML class diagram derived from the system's documentation, is presented in **Figure 2**. It includes versioning (new MappingInstances for a single Mapping that must not overlap in time), attribution (the Creator class), and actions (Actions) that are conditional on URI pattern matches, HTTP variable matches, or HTTP header values (Conditions). A snippet of the XML exchange format for a Mapping instance, implemented to allow both identifier data transfer and data migration away from a particular PID Service version or even away from the PID Service itself, is given in **Figure 3**. This more detailed PID data model is an approximation of the of the data model used by proxy servers for URI redirection such as Apache. However, unlike with the use of Apache server, non-HTTP-based information may not be used by the PID Service (Apache server may base redirection on the existence of particular files on the server). Additionally, the PID Service has some features that are difficult to implement in Apache server: strict rule hierarchy and massive scalability of patterns/rules.

The PIMs for HTTP PIDs in **Figures 1, 2** and **3** do contain protocol-independent metadata, such as creation date (**Figure 1b**) and Creator (**Figure 2**) however for resolution, they are all bound to the HTTP specification (Fielding *et al.* 2016) which is expressed as an HTTP redirect with an HTTP Status Code.

id	path	type	target	created	last_modified	status	indexed
1	/example/path	302	http://example.com/redirectedPath	2016-02-29 T13:08:11	2016-02-29 T14:08:11	OK	1
2	/example/path/deeper	302	http://example.com/redirectedDeeper	2016-02-29 T13:09:11	2016-02-29 T14:09:11	OK	1

Table 2: Examples of data for the class model in Figure 1b (Soiland-Reyes, 2016b).

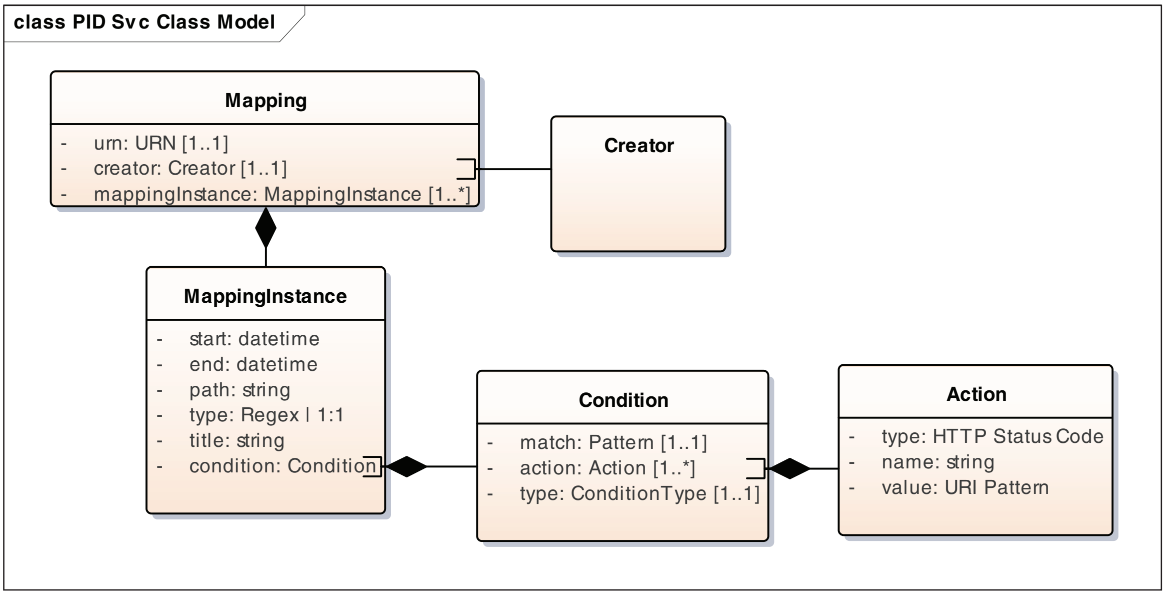
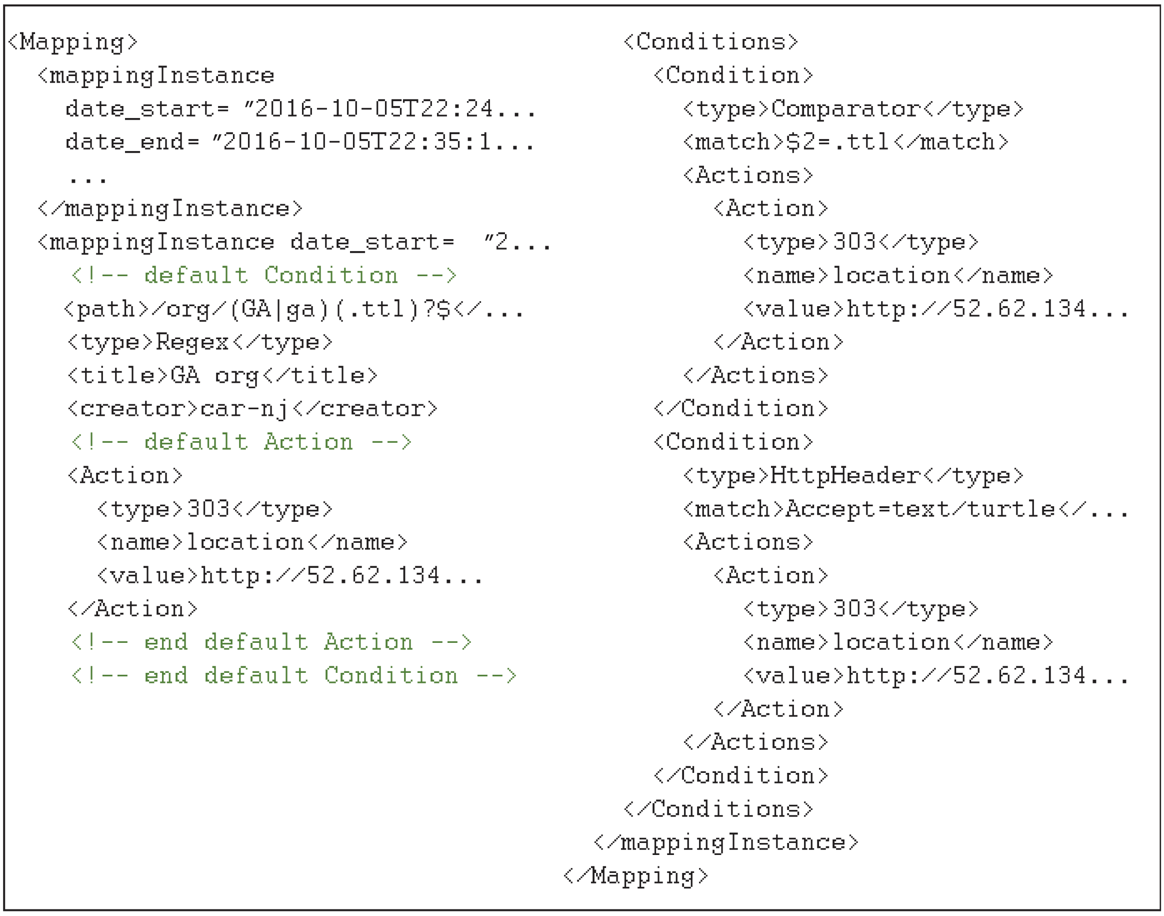


Figure 2: Part of the HTTP URI PIM implemented by the PID Service. The 'URN' object is a Universal Resource Name (Moats, 1997) and the MappingInstance 'type' property has a shorthand notation indicating allowed values of either 'Regex' or '1-to-1'. The Mapping to Mapping instance relationships which enable a strict Mapping hierarchy are not shown.



```
<Mapping>
  <mappingInstance
    date_start= "2016-10-05T22:24...
    date_end= "2016-10-05T22:35:1...
    ...
  </mappingInstance>
  <mappingInstance date_start=  "2...
    <!-- default Condition -->
    <path>/org/(GA|ga)(.ttl)?$</...
    <type>Regex</type>
    <title>GA org</title>
    <creator>car-nj</creator>
    <!-- default Action -->
    <Action>
      <type>303</type>
      <name>location</name>
      <value>http://52.62.134...
    </Action>
    <!-- end default Action -->
    <!-- end default Condition -->
    <Conditions>
      <Condition>
        <type>Comparator</type>
        <match>$2=.ttl</match>
        <Actions>
          <Action>
            <type>303</type>
            <name>location</name>
            <value>http://52.62.134...
          </Action>
        </Actions>
      </Condition>
      <Condition>
        <type>HttpHeader</type>
        <match>Accept=text/turtle</...
        <Actions>
          <Action>
            <type>303</type>
            <name>location</name>
            <value>http://52.62.134...
          </Action>
        </Actions>
      </Condition>
    </Conditions>
  </mappingInstance>
</Mapping>
```

Figure 3: An XML serialised instance of the PID Service's PID PIM for a Mapping. Shown are the default actions (redirection to an HTML page) as well as pattern-based conditional redirects for a pseudo file extension (.ttl) and an HTTP Accept header set to the MIME type text/turtle, both of which redirect to Turtle RDF serialisations (W3C, 2014) of the same resource presented in the HTML default case. 'Turtle' is a W3C recommendation for serialialising Resource Description Framework resources.

A new PID PIM

We introduce a new Platform Independent Model for PID metadata to better cater for some aspects of the pillars mentioned above. **Figure 4** shows a class model of the PIM.

For *Identifier Independence* pillar, the recommendation to define which, if any, pattern matching system is used is catered for by properties of the PersistentIdentifier class: pattern and patternType.

For the *Issuing Identifiers* part of the *Delivering Essential PID Functions* pillar, the recommendation to ensure that PID ownership is known is catered for by associating the PersistentIdentifier class with, at least, a Custodian and a Publisher (we take the role definitions for these two agents to be those of the same name in the ISO19115 standard's codelist (ISO, 2008)) with the Publisher operating also as the fallback for the identifier in the case of abandonment by the Custodian.

For *Storing Identifiers*, the recommendation to ensure metadata interpretability could be catered for by adherence to a PID PIM with well-defined elements, therefore in this model we have reused property names from other, well known, schema and we are assisted in just publishing the model itself.

Also for *Storing Identifiers*, the recommendation to catering for versioning and provenance is addressed by properties of the PersistentIdentifer class: wasRevisionOf and hadRevision point to previous and future versions of this PID and generatedAtTime and invalidatedAtTime tie down the time bounds for the creation and deprecation of this PID. These four property names are taken from the standardised provenance data model, PROV-DM (Moreau & Missier 2013), in order to maximise interpretability, as discussed above.

For this generic PIM, no details can be given as to the properties required of the Condition and Target classes for PID operation. The Condition and Action classes shown in **Figure 2** can be taken as HTTP-specific implementations. Furthermore, no details can be given for the Protocol and PatternType classes, other than that they should be presented in order to indicate which Protocols and PatternType (if any) is used.

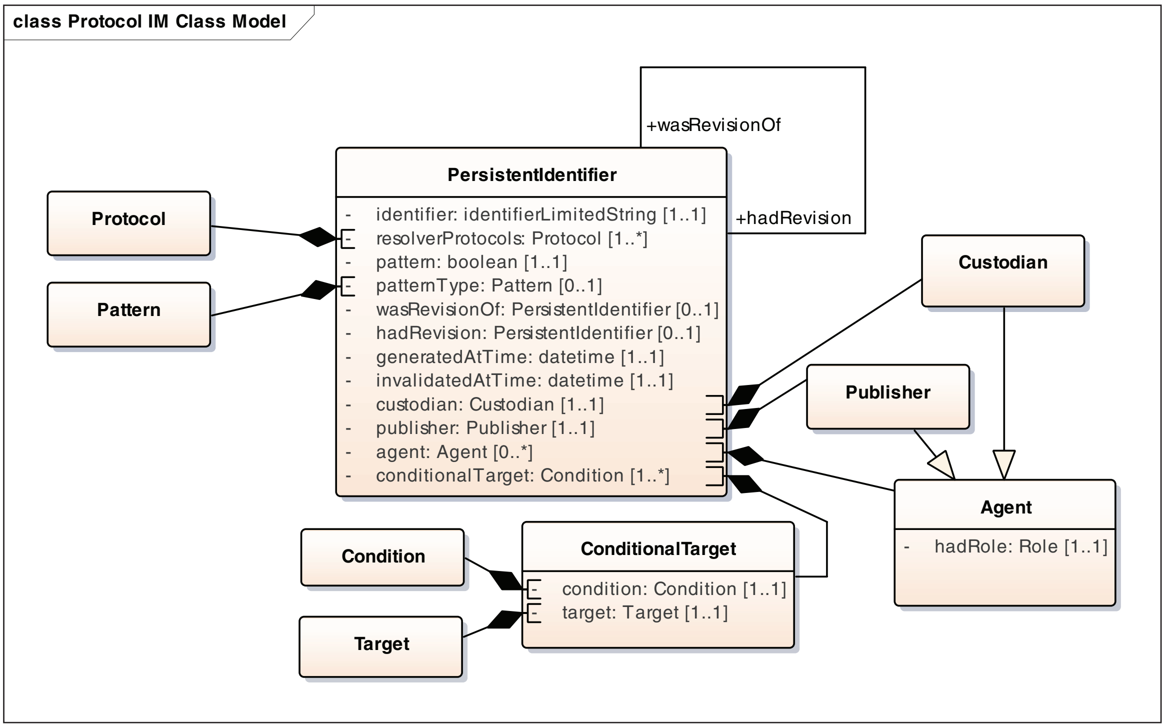

class Protocol IM Class Model
PersistentIdentifier
- identifier: identifierLimitedString [1..1]
- resolverProtocols: Protocol [1..*]
- pattern: boolean [1..1]
- patternType: Pattern [0..1]
- wasRevisionOf: PersistentIdentifier [0..1]
- hadRevision: PersistentIdentifier [0..1]
- generatedAtTime: datetime [1..1]
- invalidatedAtTime: datetime [1..1]
- custodian: Custodian [1..1]
- publisher: Publisher [1..1]
- agent: Agent [0..*]
- conditionalTarget: Condition [1..*]
+wasRevisionOf
+hadRevision
Protocol
Pattern
Custodian
Publisher
Agent
- hadRole: Role [1..1]
Condition
Target
ConditionalTarget
- condition: Condition [1..1]
- target: Target [1..1]

Figure 4: A proposed Platform Independent Model of a PID's metadata.

Implementation Case Study

Geoscience Australia (GA) is an Australian Government agency responsible for geoscientific and spatial information. The agency houses many data collections — digital, paper and physical samples — and uses a range of identifiers for different catalogue items. DOIs are used for public digital datasets (e.g., doi:10.4225/25/5823c37333f9d), ISBNs are used for reports in series (e.g., 978-1-921954-52-8; Hitchman *et al.* 2011), URIs for digital datasets without DOIs (e.g., http://pid.geoscience.gov.au/dataset/69674) and international GeoSample Numbers (IGSNs) (IGSN e.V.) for samples (e.g., igsn:10273/AU239).

Object diagrams showing implementations of the PID PIM in **Figure 4** are given for two of the above-quoted identifiers in **Figures 5–6**. In **Figure 5** and **Figure 6**, the PIM is sensibly implemented, despite differences in the metadata needed for the functioning of IGSNs and HTTP URIs. **Figure 5** shows the IGSN PID metadata relating to a previous version – perhaps the Target of the IGSN was changed. **Figure 6** shows two different ConditionalTargets for the HTTP URI PID: the first is the default, no conditions, resolving to an HTML catalogue web page Target. The second is conditional on an HTTP Request Accept header prioritising an RDF serialisation of the metadata, leading to a Turtle file Target.

These instances of the PID PIM are implemented by the Publisher of the PIDs, GA, and not by the international PID systems (the IGSN network and global DNS for the IGSN and HTTP URI instances respectively) with which they operate. It would be preferable for those systems to be the implementers, or at least for there to be a distributed set of implementers rather than one agency, but this is not yet possible with a new model.

The PID PIM instances just described contain information that will assist in the event that either the Custodian or Publisher wishes to make a change to the PID or that some external driver, such as technology change, causes it. For example, if IGSNs cease to resolve due to the future unavailability of the IGSN resolver network (IGSNs use Handle), GA, the Distributor of IGSNs and other PIDs, could identify PIDs affected and migrate them, perhaps to HTTP URIs. Geoscience Australia actually maintains both Handle and HTTP URI resolution of sample identifiers as a guard against this sort of scenario and that is indicated in **Figure 5** by the presence of two Protocol instances.

The implementation of the PID PIM at GA assists with the implementation of policy as recommended in the *Employing Policies for Change* pillar. In addition to the explicit nomination of Custodians assisting with *Identifier Abandonment*, the existence of the model itself assists with generating policy to handle technology change and will eventually assist with decommissioning if (when) GA no longer requires identifiers. As

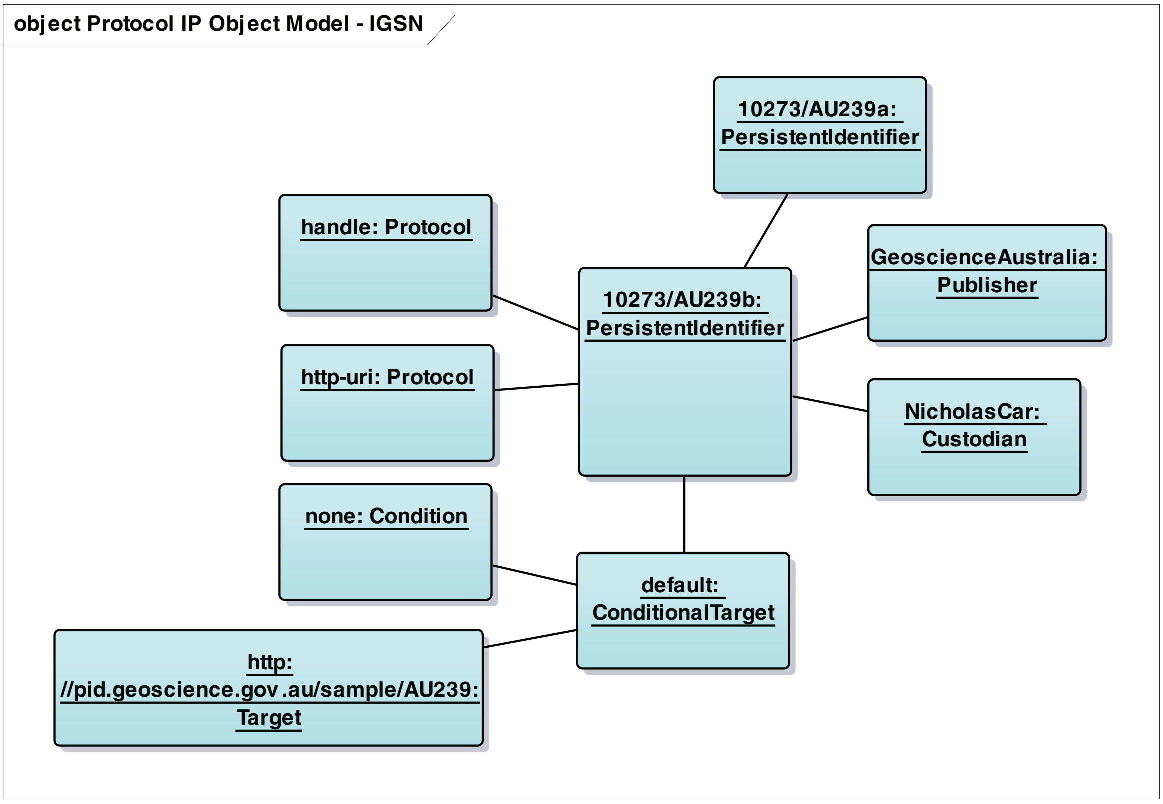

object Protocol IP Object Model - IGSN
10273/AU239a: PersistentIdentifier
handle: Protocol
10273/AU239b: PersistentIdentifier
GeoscienceAustralia: Publisher
http-uri: Protocol
NicholasCar: Custodian
none: Condition
default: ConditionalTarget
http: //pid.geoscience.gov.au/sample/AU239: Target

Figure 5: Object Model of the PID PIM for a Geoscience Australia IGSN identifier (igsn:10273/AU239).

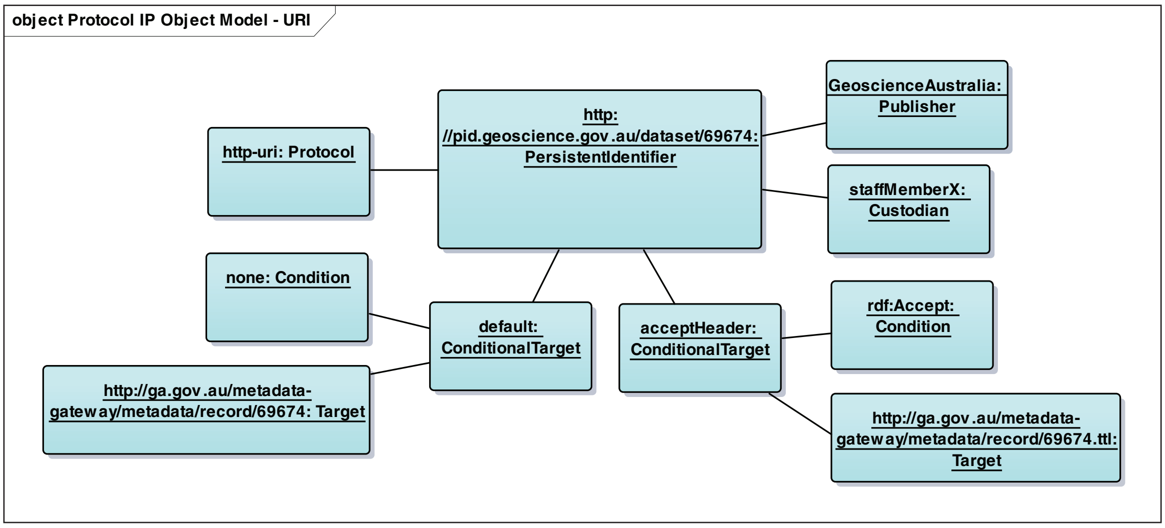

object Protocol IP Object Model - URI
http-uri: Protocol
http: //pid.geoscience.gov.au/dataset/69674: PersistentIdentifier
GeoscienceAustralia: Publisher
staffMemberX: Custodian
none: Condition
default: ConditionalTarget
acceptHeader: ConditionalTarget
rdf:Accept: Condition
http://ga.gov.au/metadata-gateway/metadata/record/69674: Target
http://ga.gov.au/metadata-gateway/metadata/record/69674.ttl: Target

Figure 6: Object Model of the PID PIM for a Geoscience Australia HTTP URI identifier (dataset no. 69674).

mentioned above, both global DOI and the IGSN consortium would benefit from adopting something like the PID PIM too.

Table 3 summarises adherence of just one of the GA systems, that of the IGSNs in **Figure 5**, to the pillars.

Future Work – an opportunity for use

Any PID system could compare itself to the Pillars and attempt to map its PID metadata to the PID PIM however we see a significant opportunity for application with W3C's Permanent Identifier Community (W3ID) Group. That group has so far neither addressed all of the aspects of the Pillars or established a PIM nor has it excluded doing so through governance or technical setup lock-in.

Pillar	PID Service Implementation
Identifier Independence	Does *Avoid organisation names* – through the implementation of a non-organisation-specific domain name. Does *Avoid technology references* – by governance of PID patterns. Does *Avoid resolution protocol indicators* – and implements two resolution protocols. Uses a User Interface that prevents *characters problematic for well-known protocols.* Does *Avoid visual ambiguity and use a well-known character set* – UTF-8. Does *Define which, if any, pattern matching system they use.*
Delivering Essential PID Functions	Does *Issue identifiers* – the implementation's primary task: *Uniqueness* – enforced locally by a lower-level system (database primary key) and globally through adherence to the IGSN community procedures. *Ownership* – recorded for every PID (fairly trivial for a single-agency implementation). *Editable* – PID metadata editable via UI & API. Does *Store identifiers* – the tool uses a corporate database: *Scalability* – potentially limitless, using a large-scale corporate database. *Integrity* – implemented with normal corporate database procedures. *Interpretability* – documented by the data model and system documentation. *Versioning* – captured and stored but only exposed to administrator users. Does *Resolve identifiers* – using two protocols.
Separation from Data Delivery	Inherent: no ability for the system to deliver data.
Employing policies for change	*Technology change* – the system has been set up with adaption to technology change in mind, as per a corporate policy at GA. As such, the resolver mechanism and the identifier data store are loosely coupled. The identifier data store is mapped to the PIM and exportable according to that data model. *Social change* – the system has an institutional owner and is thus able to handle individual staffing changes (Custodians). Publisher change catered for to some extent by participation of GA in the IGSN network meaning that copies of the identifiers and a limited version of their metadata are replicated to other organisations (such as CSIRO, two of the authors' organisation). *Identifier abandonment* – identity of each identifier's Custodian is captured. System admin has access to all. *Financial sustainability* – not addressed. The implementation assumes an on-going institutional budget both for internal system management and adherence to the IGSN community. This is a major weakness of this system. *Decommissioning* – documented and comprehensive export formats (the XML shown in **Figure 3**) assist with this and change procedures are in place for each element of the system in accordance with corporate systems policy.

Table 3: The extent to which the identifier system in Figure 5 at Geosciences Australia implements the four Pillars.

The group's aim is to "setup and maintain a secure, permanent URL re-direction service for the Web" (Sporny 2013). The group differentiated itself from PURLs on technical grounds (lack of features) and organisational grounds (lack of a "fail-over plan"). In late 2015 and until mid-2016, observing the decline of the PURL system and before its management by the Internet Archive (Internet Archive 2016), the W3ID group discussed porting the PURL identifier data and resolution system to the W3ID system. Previous to that, discussion about the technical and organisational nature of the W3ID group's system was had with emphasis being placed on technology-independent URI metadata (see Sporny 2013 and related emails). Due to these discussions and system setup, such as multiple administrators, data replication and platform-independent tooling, some aspects of the Pillars are catered for, but not all. Of particular concern to these authors are:

- *Scalability* – all of the identifiers are stored in a Git-based (SFC 2017) version control repository. This may not cater for many millions of identifiers or identifier patterns;
- *Social change* – a very few individuals run the system and it's not clear what would happen, if they ceased involvement or if conflict arose between them. While Custodians are known, the handling of abandoned identifiers is not defined (only that "those of interest will be maintained");

- *Financial sustainability* – a difficult issue for many PID systems (such as those described above), it is not addressed by the W3ID group;
- *Decommissioning* – even though well-known, tested, systems are used, there is no explicit plan for how to decommission them other than perhaps an assumed direct export of the system data (text files using Apache's mod_rewrite (Apache 2017)).

This is a very superficial assessment of the group and the system. A more complete assessment of the W3ID group's system's technical composition and governance could be conducted and such an exercise may either validate or invalidate the Pillars or PIM design or prompt the group to consider their adoption.

The opportunity here is not for an assessment of the value of the W3ID group's work to the community – time alone will tell as the system is relatively new (3 years old) – however if the W3ID group were not to consider all the Pillars, we may see another PURL near-death experience in a decade or two, even if the system becomes popular.

Conclusion

In this paper, we have explored the trustworthiness of persistent identifier (PID) systems and derived a set of recommendations (the Pillars) for the design of a trustworthy PID. The proposed recommendations cover various aspects of a PID system from the governance and policymaking to the definition of critical components that deliver essential PID system functions and avoiding technology, and even protocol-specific PIDs. We have placed a strong emphasis on separation of data delivery from PID management and have treated PIDs as data items themselves. We have also separated PID resolution from PID metadata management as we acknowledge resolution systems may change over time too.

To progress the maturity of knowledge around PID metadata, we have introduced a Platform Independent Model (PIM) that captures the critical metadata needed for the management of PIDs, regardless of the data they identify and how they are resolved. We have shown two nascent implementations of the PID PIM in order to test its applicability. We find that the PID PIM, in its current form, certainly covers a range of PID scenarios, but accepts that only through further, 3rd party, implementation tests will we see how comprehensively it covers PID requirements.

With the recommendations presented and the tool of the PID PIM, we believe we have added to the thought that could be applied to the design of new PID systems in order to avoid some of the pitfalls of systems that were designed to last 'forever' but which after only several decades are dormant or have closely escaped death.

Competing Interests

The authors have no competing interests to declare.

References

Apache Software Foundation, The (Apache) 2017 Redirecting and Remapping with mod_rewrite. Web page. Available at: https://httpd.apache.org/docs/2.4/rewrite/remapping.html (Last accessed 15 January 2017).

Atkinson, R and **Box, P** 2016 Spatial Identifier Reference Framework – spatial linked data description, configurations and XSL rendering. Version 1. CSIRO Data Collection. DOI: https://doi.org/10.4225/08/571EF7060EEAF

Atlas of Living Australia (ALA) 2017 About the Atlas. Web page. Available at: http://www.ala.org.au/about-the-atlas/ (Last accessed 15 January 2017).

Barnett, B 2011 Regular Expressions. Web page. Available at: http://www.grymoire.com/Unix/Regular.html (Last accessed 15 January 2017).

Beck, K, Ritz, R and **Wittenburg, P** 2016 Towards a Global Digital Object Cloud – Report from the Views on PID Systems training course and workshop. In: *RDA Europe Workshop, Max Planck Compute and Data Facility (MPCDF)*. Garching-Munich, Germany.

Berners-Lee, T 1998 Cool URIs don't change. Available at: https://www.w3.org/Provider/Style/URI.html (Last accessed 26 October 2016).

Bilder, G, Lin, J and **Neylon, C** 2015 Principles for Open Scholarly Infrastructure-v1. DOI: https://doi.org/10.6084/m9.figshare.1314859 (Last accessed 15 January 2017).

Buringh, E and **Van Zanden, J L** 2009 Charting the "Rise of the West": Manuscripts and Printed Books in Europe, A Long-Term Perspective from the Sixth through Eighteenth Centuries. *Journal of Economic History*, 69(2). DOI: https://doi.org/10.1017/S0022050709000837

Bütikofer, N 2009 Catalogue of criteria for assessing the trustworthiness of PI systems, nestor-Materialien, Niedersächsische Staats und Universitätsbibliothek Göttingen, Göttingen, Germany. Available at: http://nbn-resolving.de/urn:nbn:de:0008-20080710227 (Last accessed 11 November 2016).

Chief Technology Officer Council (CTOC) 2011 Designing URI Sets for Location. In: The report from the Public Sector Information Domain of the CTO Council's cross Government Enterprise Architecture, and the UK Location Council. Available at: https://data.gov.uk/library/designing-uri-sets-for-location (Last accessed 3 November 2016).

Corporation for National Research Initiatives (CNRI) 2015 HANDLE.NET®: Technical Manual Version 8.1 Preliminary edition. HDL:20.1000/105.

Corporation for National Research Initiatives (CNRI) 2016 HDL.NET Information Services. Web page. Available at: http://handle.net/ (Last accessed 15 January 2017).

CSIRO 2017 Data Access Portal. Web site. Available at: https://data.csiro.au (Last accessed 15 January 2017).

DataCite Metadata Working Group (DMWG) 2014 DataCite Metadata Schema Documentation for the Publication and Citation of Research Data. Version 3.1. DataCite e.V. DOI: https://doi.org/10.5438/0010

Dewey, M 1876 A Classification and Subject Index for Cataloguing and Arranging the Books and Pamphlets of a Library. Amherst Massachusetts, USA. Available at: http://www.gutenberg.org/ebooks/12513 (Last accessed 26 October 2016).

Farrell, S, Kutscher, D, Dannewitz, C, Ohlman, B, Keranen, A and **Hallam-Baker, P** 2013 Naming Things with Hashes. *RFC 6920*. DOI: https://doi.org/10.17487/rfc6920

Fielding, R, Gettys, J, Mogul, J, Frystyk, H, Masinter, L, Leach, P and **Berners-Lee, T** 1999 RFC2616: Hypertext Transfer Protocol – HTTP/1.1. Network Working Group of the Internet Engineering Taskforce. Available at: https://www.ietf.org/rfc/rfc2616.txt (Last accessed 26 October 2016).

Finkel, I and **Taylor, J** 2015 cuneiform. London, UK. The British Museum Press. ISBN: 978-0-7141-118-9.

Golodoniuc, P 2015a Persistent Identifier Service (PID Service). Wiki web page. Available at https://www.seegrid.csiro.au/wiki/Siss/PIDService [Last accessed 3 November 2016].

Golodoniuc, P, Car, N J, Cox, S J D and **Atkinson, R A** 2015b PID Service – an advanced persistent identifier management service for the Semantic Web. In: *The 21st International Congress on Modelling and Simulation (MODSIM2015)*. Modelling and Simulation Society of Australia and New Zealand, Broadbeach, Australia, December 2015, pp. 767–773. ISBN: 978-0-9872143-5-5. Available at: http://mssanz.org.au/modsim2015/C8/golodoniuc.pdf (Last accessed 26 October 2016).

Golodoniuc, P, Car, N J and **Klump, J** 2017 Distributed persistent identifiers system design. *Data Science Journal* (this volume).

Golodoniuc, P, Klump, J and **Car, N J** 2016 Trustworthy persistent identifier systems of the future. *Geophysical Research Abstracts*, 18: EGU2016-1506-2, Copernicus Society. Available at: http://meetingorganizer.copernicus.org/EGU2016/EGU2016-1506-2.pdf (Last accessed 26 October 2016).

Graham, M 2016 Persistent URL Service, purl.org, Now Run by the Internet Archive. Available at: https://blog.archive.org/2016/09/27/persistent-url-service-purl-org-now-run-by-the-internet-archive/ (Last accessed on 26 October 2016).

Hazel, P 2012 PCRE – Perl-compatible regular expressions (original API). Library Functions Manual. Available at: http://pcre.org/pcre.txt (Last accessed 15 January 2017).

Hitchman, A P, Crosthwaite, P G, Jones, W V, Lewis, A M and **Wang, L** 2011 Australian Geomagnetism Report 2010, Geoscience Australia Record 2011/41. Geoscience Australia.

Huber, R and **Klump, J** 2016 How dead is dead in the PID Zombie Zoo? In: *RDA Europe Workshop, Max Planck Compute and Data Facility (MPCDF)*. Garching-Munich, Germany.

IGSN e.V 2016 IGSN e.V.: International GeoSample Number. Web site. Available at: http://www.igsn.org (Last accessed 15 January 2017).

International DOI Foundation 2016 Factsheet: Key Facts on Digital Object Identifier System. Corporate news web page. Available at: http://www.doi.org/factsheets/DOIKeyFacts.html (Last accessed 15 January 2017).

International Working Group on Taxonomic Databases 2017 LSID Resolution Project. Web page. Available at: http://www.lsid.info/ (Last accessed 15 January 2017).

Internet Archive 2016 What is a PURL? Corporate web page. Available at: https://archive.org/services/purl/help [Last accessed 15 January 2017].

ISO 2008 gmxCodelists – XML Codelists for description of metadata datasets compliant with ISO/TC 211 19115:2003 and 19139. XML document available at: http://www.isotc211.org/2005/resources/Codelist/gmxCodelists.xml (Last accessed 26 October 2016).

ISO 2012 Information technology – Universal Coded Character Set (UCS). ISO/IEC Standard 10646:2012.

King, B 2011 Too Much Content: A World of Exponential Information Growth. Huffington Post blog article. Huffington Post blog post. Available at: http://www.huffingtonpost.com/brett-king/too-much-content-a-world-_b_809677.html (Last accessed 15 January 2017).

Kuny, T 1998 The Digital Dark Ages? Challenges in the Preservation of Electronic Information. Proc. Of the 63RD IFLA Council and General Conference. Available at: http://archive.ifla.org/IV/ifla63/63kuny1.pdf (Last accessed 15 January 2017).

Larsen, P O and **von Ins, M** 2010 The rate of growth in scientific publication and the decline in coverage provided by Science Citation Index. *Scientometrics*, 84(3). DOI: https://doi.org/10.1007/s11192-010-0202-z

Mendelsohn, N 2006 My conversation with Sean Martin about LSIDs, public email to W3C TAG mailing list 25-Jul-2006. Available at: http://lists.w3.org/Archives/Public/www-tag/2006Jul/0041 (Last accessed 26 October 2016).

Moats, R 1997 URN Syntax. Internet Engineering Taskforce Request for Comments: 2124. Available at: https://www.ietf.org/rfc/rfc2141.txt (Last accessed 15 January 2017).

Mockapetris, P 1987 Domain Names – Concepts and Facilities. Internet Engineering Taskforce Request for Comments: 1034. Available at: https://tools.ietf.org/rfc/rfc1034 (Last accessed 15 January 2017).

Mohr, G 2002 MAGNET-URI Project. Web page. Available at: http://magnet-uri.sourceforge.net/ (Last accessed 15 January 2017).

Moreau, L and **Missier, P** (eds.) 2013 PROV-DM: The PROV Data Model. W3C Recommendation. Available at: https://www.w3.org/TR/prov-dm/ (Last accessed 26 October 2016).

Newman, L H 2016 What We Know About Friday's Massive East Coast Internet Outage. *Wired Magazine*, 21 October 2016. Available at: https://www.wired.com/2016/10/internet-outage-ddos-dns-dyn/ (Last accessed 28 October 2016).

Object Management Group (OMG) 2004 Life Sciences Identifiers Specification. OMG Report dtc/04-05-01. Available at: http://www.omg.org/cgi-bin/doc?dtc/04-05-01 (Last accessed 15 January 2017).

Online Computer Library Center, Inc. (OCLC) 2016 OCLC and Internet Archive work together to ensure future sustainability of Persistent URLs. Corporate website news article. Available at: https://www.oclc.org/en/news/releases/2016/201623dublin.html (Last accessed 15 January 2017).

R.R. Bowker LLC 2014 About the ISBN Standard. Web page. Available at: http://www.isbn.org/about_ISBN_standard (Last accessed 15 January 2017).

Ribeiro, J 2016 U.S. says transfer of internet governance will go ahead on Oct 1. *Computerworld*, 17 August. Available at: http://www.computerworld.com/article/3108617/internet/us-says-transfer-of-internet-governance-will-go-ahead-on-oct-1.html (Last accessed 26 October 2016).

Soiland-Reyes, S 2016a Stian's Gists: rewrite.csv. GitHub example code. Available at: https://gist.github.com/stain/c2d668b11b66948b5991 (Last accessed 26 October 2016).

Soiland-Reyes, S 2016b w3id-csv: purl_example.csv. GitHub example code. Available at: https://github.com/stain/w3id-csv/blob/master/purl_example.csv (Last accessed 26 October 2016).

Sporny, M 2013 [via Permanent Identifier Community Group]. Email to a mailing list. Archived and available online at: https://lists.w3.org/Archives/Public/public-perma-id/2013Feb/0000.html and https://lists.w3.org/Archives/Public/public-perma-id/2013May/0000.html (Last accessed 15 January 2017).

Software Freedom Conservancy (SFC) 2017 Git. Web site. Available at: https://git-scm.com (Last accessed 15 January 2017).

Stadd, A 2013 Understanding The Growth Of Information. Social Time/Adweek web page. Available at: http://www.adweek.com/socialtimes/growth-of-information/477932 (Last accessed 15 January 2017).

World Wide Web Consortium (W3C) 2014 RDF 1.1 Turtle: Terse RDF Triple Language. W3C Recommendation 25 February 2014. Available at: https://www.w3.org/TR/turtle/ (Last accessed 15 January 2017).

Zittrain, J L, Albert, K and **Lessig, L** 2013 Perma: Scoping and Addressing the Problem of Link and Reference Rot in Legal Citations. Harvard Public Law Working Paper no. 13–42. DOI: https://doi.org/10.2139/ssrn.2329161

How to cite this article: Car, N J, Golodoniuc, P and Klump, J 2017 The Challenge of Ensuring Persistency of Identifier Systems in the World of Ever-Changing Technology. *Data Science Journal,* 16: 13, pp. 1–18, DOI: https://doi.org/10.5334/dsj-2017-013

Submitted: 16 November 2016 **Accepted:** 10 March 2017 **Published:** 04 April 2017

]u[*Data Science Journal* is a peer-reviewed open access journal published by Ubiquity Press.

OPEN ACCESS