Towards Electronic Persistence Using ARK Identifiers

John A. Kunze

California Digital Library
University of California, Office of the President, Oakland, CA 94612, USA
jak@ucop.edu

Abstract. The ARK (Archival Resource Key) is the only persistent naming scheme for internet information objects that squarely faces the service issues underlying practical electronic permanence. The ARK is a specially constructed, actionable URL encapsulating a globally unique identity that is independent of the current service provider. Each ARK is by definition bound to three things: (a) object access, (b) object metadata, and (c) a faceted commitment statement. These service bindings provide the best way for service providers to communicate their varied commitments to objects. It is for these reasons that the California Digital Library is pursuing the use of ARKs.

1 Introduction

Web users know the irritation of broken URLs [2]. Returning to a web address only to find the desired object missing is more than an irritation for scholars and catalogers, whose trades rely on persistent reference. If there is one thing that distinguishes a digital library from a mere web site, it is that libraries do their best to provide reliable, persistent access through durable links. To that end much discussion and development activity has focused on persistent identifiers, in other words, on links that continue to provide access into the indefinite future. The popular vision of ubiquitous, technologically-hardened identifiers has a deceptively simple problem statement:

1. The goal: long-term identifiers.
2. The problem: URLs break.
3. Why URLs break: because objects are moved, removed, and replaced.
4. Conventional hypothesis: use indirect names (PURLs, URNs, Handles) instead of URLs; what worked for the DNS (Domain Name System) [11] in stabilizing internet hostnames should work for digital object references.

The conventional approach has yielded no convincing solutions, in part because of problem statement flaws that this paper will expose and correct. The three best known persistent identifier schemes come from this approach. In the early 1990's, the design of the URN (Uniform Resource Name) [10] responded

to the observed failure rate of URLs by articulating an indirect, non-hostname-based naming scheme and the need for responsible name management. Meanwhile, promoters of the DOI (Digital Object Identifier) [6] succeeded in building a community of providers around a mature software system that implements the Handle [9] naming scheme. The PURL (Persistent Uniform Resource Locator) [15] was another scheme that had the unique advantage of working with unmodified web browsers.

The insights contributed by these schemes joined with those from other proposals to form an incomplete patchwork of strategies. Replication, for example, is a strategy employed by many repositories (e.g., the Internet Archive [7]) and formalized in systems such as LOCKSS [13] and SRB [1]. Other proposals such as [12] involve content-based computations designed to strengthen the binding between names and objects. Proposals dealing with repository accreditation [5] and online reputation building [14] are relevant to socio/cultural strategies.

Representing a new approach is the ARK (Archival Resource Key) scheme. A founding principle of the ARK is that persistence is purely a matter of service, and is neither inherent in an object nor conferred on it by a particular naming syntax. The best an identifier can do is lead users to those services.

2 Conventional Indirection: Influence of DNS

The conventional hypothesis suggests that much of the problem can be solved by using indirect names – names that are resolved to URLs (presumably the best and most current URLs) at the moment of access. Perhaps what makes this approach so appealing (aside from the glib axiom that in information science there is no problem that cannot be solved by adding one more level of indirection) is the spectacular success and elegance of the internet DNS (Domain Name System) [11] in solving what might appear to be a similar problem. DNS was so effective at using tables of indirect names to stabilize how we refer to internet hosts (e.g., 128.218.39.5 before DNS and www.lib.ucsf.edu with DNS), that it seemed natural to follow its lead for persistent object identifiers.

The flaw in this hypothesis, however, has to do with an inversion in the perceived value of the named things. An incorrect DNS table entry for a hostname that results in the disappearance from the network of a costly, tangible, and unique piece of multi-purpose infrastructure (namely, a computer and its files) will be the subject of swift repair because of the high value placed on that computer's availability. On the other hand, while a high value has obviously been placed on persistent object identifiers in general, for each individual object formerly at the end of a link that remains broken for any length of time, the perceived value must be low.

Several explanations for devaluation of individual digital objects are possible, and each touches on aspects that apply more to named digital objects than to named computers. First, the sheer number of objects is very high compared to the number of computers. Second, many a digital object, if it has any current value, exists elsewhere in copies, such as on a bookstore or library shelf, on a

mirrored web site (publicly valued items), or in a privately held collection (e.g., on one's home computer). Such an object is not unique and its disappearance may not actually have been inconvenient enough to be worth complaining about if a suitable alternate copy could be located quickly; in a sense, the higher the initial value of the document, the more its perceived value is likely to decrease. Digital objects are also much less tangible – hence much easier to lose, forget about, and devalue – than computer equipment. Finally, the value of a shared computer tends to be stable during its operational lifetime, but the value of many information objects is high only during discrete moments. For example, if the time between initial obscurity for a published work and its eventual discovery is small enough, there is more of a chance that a copy will still exist somewhere.

Although DNS has much to teach in the area of name management, given the amount of non-overlap in solution activities, it is not at all clear that what will work for management of computer hostnames will also work for management of information objects.

3 Terminology

The main trouble with the previous problem statement is ambiguous terminology. Starting with basic terms, if one is looking literally for *persistent identifiers*, they already exist in schemes such as ISBN and LCCN. It turns out that what people really want is the kind of convenient access to which they have become accustomed on the Web. Instead, what we're looking for are persistent *actionable* identifiers, where an *actionable* identifier is one that widely available tools such as web browsers can use to convert a simple "click" into access to the object or to a doorway leading directly to access (e.g., the doorway could be a password challenge or object summary description page).

As a consequence, it makes little sense to design for persistent identification without designing for persistent access at the same time. For even if a perfect system for assigning forever unique identifiers were created, if it did so without reducing access failure rates, no one would be interested. The central issue – which may be summed up as the "HTTP 404 Not Found" problem – would not have been addressed. Unfortunately, decoupling persistent identification from access was an early decision made by designers of PURLs, URNs, and Handles.

Another terminological failing has caused the misapprehension that URLs are unsuitable for persistent identification because they can appear to contain "locations" or "addresses". Most users are completely unaware of the many routing table, proxy server, web server, and operating system indirection operations required to resolve URLs. Moreover, without access to logs from the entire resolution system, not even an expert can say with certainty where a URL-identified object is located. This is not to say that some naming practices still common today don't create URLs that contain clues reflecting location information, just that URLs may be just as indirect and location-free as any indirect identifier (e.g., URN or Handle).

Another foundation term is *identifier* itself. Commonly defined [2] on the Web as a "string", it's hard to talk about an identifier that "breaks." When does a string "break," except perhaps when it sustains damage to its character sequence? What really breaks has to do with the identifier's reference role, which leads us to a better definition. An identifier is an *association* between a string and an information object. For the association to be verifiable, that association is made manifest by a record (e.g., a cataloging or other metadata record) that binds the identifier string to a set of identifying object characteristics.

The binding would be much easier to verify if the object carried its own identifier as part of itself (e.g., imprinted on the first page). In general, however, the identifier (the association) must be vouched for by some sort of record. Otherwise, a would-be identifier string is meaningless data, even if it were, for example, a well-formed and plausible URL. In any case, such a record is needed because objects are often too large or too encumbered to inspect conveniently (such as when licensing restrictions are in effect).

4 Setting User Expectations

For URLs that are not intended to break, where does that intention reside and how is it communicated to users? Conventionally, a literal marker string such as `purl`, `doi:`, `urn:`, or `ark:` indicates some suggestion of a promise to the recipient. One problem with conventional approaches is that a persistence promise is not a simple binary (yes/no) statement; rather, it is a faceted statement [3], including dimensions such as (1) how long the identifier (association) will be valid, (2) how long the object (access) will be available, and (3) how invariant its content is (e.g., subject to correction only, or to revision)

Once a promise is conceived, the service provider must be able to convey it to users. That promise binds the word of an identified steward to a specific set of responsibilities. No one can tell if successful stewardship will take place because no one can predict the future. Reasonable conjecture, however, may be based on past performance. There must be a way to tie a promise of persistence to a provider's demonstrated or perceived ability – its reputation – in that arena. This is perhaps the best way we have for gauging the strength of any persistence promise. Unfortunately, conventional approaches do not support the communication of such promises, so today's users have no way to tell durable identifiers – whether URNs, DOIs, or URLs – from fragile identifiers.

The situation is further complicated because each object's persistence is a function of the actions of two classes of persistence service provider. The initial assignment of an identifier is made by a *name assigning authority* (NAA), and ongoing access to objects and metadata is provided by a *name mapping authority* (NMA). The NAA has policies about name assignment, and the NMA has policies about things like object access and content invariance. There may be several competing or cooperating NMAs serving the same identifier at once, each with a different promise. While initially the assigning and mapping activities for an object may be performed by the same organization, over decades the original

intention of the NAA is irrelevant to the intentions of the current set of NMAs. Conventional schemes model the idea of an undifferentiated name authority, but do not formalize the two vitally distinct classes of roles that it plays.

URLs don't break on their own. They are broken by providers that fail in motivation or forethought either to select a URL hostname that ages reasonably well, or to update their URL redirection tables (a built-in server feature available since the early 1990's). An equivalent amount of table-updating labor is required to maintain any naming scheme that relies on indirection, with more labor implied if indirection is required of every identifier. If an organization doesn't bother to redirect URLs now (a relatively easy and well-understood process), it is unlikely to suddenly become the kind of organization that does bother to redirect URNs or Handles (a relatively new and poorly-understood process).

5 A Revised Problem Statement

Most providers neglect URL redirection tables now because they don't care or can't afford persistence. Persistence is purely a matter of service, and a persistent identifier's job is to link to basic persistence services. This leads us to a revision of the problem statement, with clarifying changes shown in italics.

1. The goal: long-term *actionable* identifiers.
   (a) *Requirement: that identifiers deliver you to objects (where feasible).*
   (b) Requirement: that identifiers deliver you to object metadata.
   (c) Desirable: each object should wear its own identifier.
   (d) Requirement: that identifiers deliver you to statements of commitment.
2. The problem: URLs break *for some objects (that is, associations between URLs and objects are not maintained), and we have no way to tell which ones will or won't break.*
3. Why URLs break: because objects are moved, removed, and replaced – *completely normal activities – and the provider in each case demonstrates insufficient commitment to update indirection tables, or to plan identifier assignment carefully. Persistence is in the mission of few organizations.*
4. Conventional hypothesis: use indirect names (PURLs, URNs, Handles) instead of URLs; what worked for DNS should work for digital object references. *Wrong. Indirection is spectacularly successful and elegant in DNS, but it's a side issue in the provision of digital object persistence.*

The reason that electronic persistence has not been solved is that the service issues have not been squarely faced. This is what the California Digital Library is pursuing, using the ARK naming scheme.

6 A New Approach: the Archival Resource Key (ARK)

The ARK scheme embodies a sober new conception of durable reference. Icarus' dream of free flight no more resembles the actual reality of human flight in the metal cylinder of an airplane than the popular vision of ubiquitous, technologically-hardened identifiers resembles how realistic peristent identification will work. Persistence is a complex and expensive undertaking, eased only here and there by technology. It should come as no surprise that no identifier scheme can assure an object's preservation as it passes from provider to provider down through the ages. An ARK, by definition, provides strong, actionable ties to three things – the object, its metadata, and a commitment statement – without which a persistence claim cannot be credible. Of course, no one can predict every contingency that might somehow overturn the strongest promise from the most reputable provider; the ARK only equips a recipient to make the best possible judgment of durability. This is as much as one can do.

The term ARK refers both to the naming scheme and to any single identifier that conforms to it. As with conventional schemes (URL, URN, Handle, PURL), assignment of an ARK does not magically make it a persistent identifier, but instead of linking the user to just one thing, an ARK links the user to three things: the object, its metadata, and the current provider's commitment statement. Also unlike conventional approaches, the ARK scheme formalizes the very different roles played by the name assigning authority (NAA) and the name mapping authority (NMA). Otherwise, an ARK is a special kind of URL. Here is an example,

```
http://ark.cdlib.org/ark:/13030/ft4w10060w
       \___________/ \__/ \___/ \________/
             |         |     |      |
             |   ARK Label   |    Name (assigned by the NAA)
             |               |
Name Mapping Authority    Name Assigning Authority
      Hostport (NMAH)      Number (NAAN)
```

In an ordinary web browser's location field, entering an ARK by itself yields the object, while entering the ARK followed simply by a '?' yields the metadata, and entering the ARK followed by '??' yields the commitment statement. These last two server probes are easy to automate, as they produce machine-readable responses according to the protocol described in [8].

The general form of an ARK is `http://NMAH/ark:/NAAN/Name`, where in the example the NMAH (Name Mapping Authority) is `ark.cdlib.org`, the NAAN (Name Assigning Authority Number) is a globally unique number identifying the NAA, `13030`, and the Name assigned by the NAA is `ft4w10060w`. The part of the ARK before the NAAN plays no part in identifying the object or in comparing ARKs for equivalence; it only serves to make the ARK actionable. The NMAH part is temporary, disposable, and replaceable. Over time the NMAH will likely stop working and have to be replaced with a currently active service provider. The NMAH is thus a kind of identity-inert, disposable

booster rocket that launches the ARK into cyber-space while allowing for limited branding. When the Web no longer exists, the core identity of the ARK is easily recovered by isolating the part of the ARK that begins with `ark:/`. The following ARKs are synonyms for but one object.

```
http://foobar.zaf.org/ark:/12025/654xz321
http://sneezy.dopey.com/ark:/12025/654xz321
ark:/12025/654xz321
```

A carefully chosen hostname in the NMAH could last for decades. If the NMAH ever fails, the ARK specification [8] describes a lookup algorithm for finding a new NMAH for the object. The algorithm is essentially a simplification of the original URN resolver discovery algorithm that uses DNS NAPTR records. A simpler alternative lookup algorithm is based on string matching against a small, mirrored text file that functions as a registry of NAANs (currently residing at the US National Library of Medicine); this is directly analogous to how old internet host tables were used for hostname lookup before DNS had been implemented. The registry file is small since it contains one entry per NAA. The Names assigned by those NAAs are tracked in databases residing with the NAAs and NMAs.

The ARK scheme can be used to assign a persistent name to almost anything. This includes digital documents, databases, software, and websites, as well as physical objects (such as books, bones, and statues) and intangible objects (vocabulary terms, chemicals, diseases, performances).

7 Persistent Identifiers at the California Digital Library

The primary reason for pursuing ARKs at the California Digital Library (CDL) is recognition that persistence is purely a matter of service, and that giving users identifiers with which to connect to objects, object metadata, and CDL commitment statements is the best way to address the service issues surrounding persistence. The CDL only assigns ARKs to objects that it owns or controls. Currently about 80,000 ARKs have been assigned at the CDL.

Although many users are interested in persistence for objects discovered through a CDL service, but that are outside of CDL control, the CDL cannot vouch for their persistence nor any commitments made by their respective providers. ARKs would therefore be inappropriate, so in this case the CDL assigns PURLs to some of those objects because the PURL software provides a convenient administrative interface for centrally updating indirect URLs.

In assigning ARKs, the CDL strives to remove all recognizable semantics. For one thing, semantics based on natural language don't travel well to other parts of the world. Secondly, the tendency to include semantics wreaks havoc with persistence that is to outlive organizational acronyms, subject classifications, and natural language semantics (e.g., what did the three letters "gay" mean in 1958, 1978, and 1998?). Today's recognized and correct attributes are tomorrow's stale

or incorrect attributes. Moreover, today's unrecognized multi-letter combination is tomorrow's unintended but offensive acronym (and in countries in another part of the world). Whether semantics are intended or not, semantic "rot" over time is a liability if persistence is the goal.

An apparent conflict of needs arises from the banishment of semantics from ARKs. Libraries have come to rely on semantically rich URLs that are commonly used to identify digital material at publishers' web sites. In the case of serials, for example, these URLs can reveal a publisher's brand in the hostname, and reveal such facts as volume number, issue number, date, and file type among URL path components. Those facts are not only valuable information about a serial, they also prove to be authoritative when the publisher's web site returns the expected serial. By happy accident, this kind of URL is both an actionable identifier and a primitive, non-standard metadata container that comes with a clever authority mechanism. Because a standard structure for binding publisher metadata to an identifier is not in common use, there is understandable resistance to the idea of giving up these ad hoc metadata semantics in exchange for protection from a potential persistence threat that might manifest itself in a decade or so.

ARKs actually accommodate both needs easily. Because of the tight binding between an ARK and its metadata, it is always easy to create bibliographies and object-level records that place semantically opaque ARKs right next to metadata that is not only fully expressive and authoritative, but that can also be refreshed automatically at any time (using the ARK followed by a '?'). Moreover, this proximity to fully expressive metadata shields the ARK scheme from internationalization pressures that are forcing the other identifier schemes (URI [4] and Handle [16]) to deal with the serious complications surrounding inclusion of language and character set indicators inside identifiers. Since the ARK scheme is designed to push meaning out of identifiers and into metadata, ARKs can get away with the simplification of using ASCII (which is universally accepted) as the underlying character set.

The CDL also takes steps to avoid accidental semantics. It restricts the ARK character set to digits and non-vowels, doesn't create identifiers with more than two non-digits in a row, generates a check character along with each ARK (which guarantees the ARK against the most common single character and transposition errors), and even uses a kind of pseudo-random sequencing to avoid "series semantics" that could reveal the order of identifier assignment.

8 Conclusion

Clearly, the argument for using ARKs in persistent identification is compelling. Serious providers have a more complex service burden than ordinary website maintainers, and ARKs smooth the way. Because the ARK is a specially constructed, actionable URL encapsulating a globally unique identity that is independent of the current service provider, ARKs work with today's browsers and with tomorrow's tools as well. The first part of the ARK URL is a kind

of identity-inert, disposable booster rocket that launches the ARK into today's cyber-space, while preserving the globally unique core identity inside it.

Unlike the usual single-value binding of one URL with one object, each ARK is by definition bound to three things: (a) object access, (b) object metadata, and (c) a faceted commitment statement. By strongly linking names to the minimal services without which no persistence claim is credible, ARKs provide the best way for multiple service providers to communicate their varied commitments to objects. It is for these reasons that the California Digital Library is pursuing the use of ARKs for digital objects that it controls.

References

1. Baru, C., Moore, R., et al: The SDSC Storage Resource Broker. Proc. CASCON'98 Conference (1998)
2. Berners-Lee, T., et al: Uniform Resource Identifiers (URI): Generic Syntax. RFC 2396 (1998)
3. Byrnes, M.,: Defining NLM's Commitment to the Permanence of Electronic Information. ARL **212** (2000) 8–9 http://www.arl.org/newsltr/212/nlm.html
4. Dürst, Martin J.: Internationalized Resource Identifiers: From Specification to Testing, 19th International Unicode Conference (2001) http://www.w3.org/2001/Talks/0912-IUC-IRI/paper.html
5. Hedstrom, M.: It's About Time: Research Challenges in Digital Archiving and Long-Term Preservation. Draft 2002-04 http://www.si.umich.edu/digarch/Report.DFt.2.doc
6. International DOI Foundation: The Digital Object Identifier (DOI) System. (2001) http://dx.doi.org/10.1000/203
7. Kahle, B.: Archiving the Internet. Scientific American (1996)
8. Kunze, J., Rodgers, R.: The ARK Persistent Identifier Scheme. Draft-06, July 2003, Work in Progress http://www.ietf.org/internet-drafts/draft-kunze-ark-06.txt
9. Lannom, L: Handle System Overview. ICSTI Forum, No. 30 (1999)
10. Moats, R.: URN Syntax. RFC 2141 (1997)
11. Mockapetris, P.: Domain Names - Concepts and Facilities. RFC 1034 (1987)
12. Phelps, T., Wilensky, R.: Robust Hyperlinks Cost Just Five Words Each. UCB//CSD-00-1091 (2000) http://www.cs.berkeley.edu/~wilensky/robust-hyperlinks.html
13. Reich, V., Rosenthal, D.: LOCKSS: A Permanent Web Publishing and Access System. D-Lib Magazine Volume 7 Number 6 (2001) http://www.dlib.org/dlib/june01/reich/06reich.html
14. Resnick, P.: Reputation Systems. Communications of the ACM, 43(12) (2000), 45–48
15. Shafer, K., et al: Introduction to Persistent Uniform Resource Locators. (1996) http://purl.oclc.org/OCLC/PURL/INET96
16. Sun, Sam X.: Internationalization of the Handle System - A Persistent Global Name Service. 12th International Unicode Conference (1998) http://www.cnri.reston.va.us/unicode-paper.ps