Internet Identifier Issues
John C Klensin
2017-07-30

A review of the IAB's call for position papers for the 2017 October workshop led me to the realization that I have written on a number of relevant subjects in the last few weeks and years, mostly in IETF contexts. Consequently, since a position paper was requested, I'm going to try to provide a perspective on what I think are the key issues from the perspective of user-facing identifiers (and claimed identifiers), rather than going into great detail on any of those issues or treating the solicitation as an outline. Some of these issues go beyond what I interpret as the scope of the solicitation but appear to me to be closely connected to it. Obviously, there are issues for what I think of as network-facing identifiers – ones that, under normal circumstances, are visible only to protocols, implementers, operators, in back end processing, etc. – but those appear to be to lie outside the scope of the solicitation and, if they are not, I trust others will cover them..

After the first, these are not listed in any particular order.

(0) The names of things versus what they are called and how identifiers are bound to objects

The IETF has tended to use the term "identifier" in very informal ways, including for a number of names and references that would not be recognized as identifiers in the communities that have studied such things, often for centuries. Those who deal with books and similar objects would find the idea of identifying a book by, e.g., a bookcase name, shelf number, and number of centimeters from the left of that shelf to be a little bit strange, perhaps bordering on silly, especially so if there were multiple copies of the book available and users didn't care which copies they obtained. And, of course, the identity of the book does not change when it is removed from the shelf or a new book is inserted to its left. The title of a book, places where it can be found, numbers reflecting when it was acquired by a particular institution, what the book is called, and how it is identified may all be different, and often are.

(1) Identifying (or naming) objects, normally digital ones, versus identifying network locations and objects with reference to their network locations.

Prior to the late 1980s, the vast majority of discussions about, and use of identifiers in the Internet context was about names for network objects. Early documents make it fairly clear that host names were seen as simply mnemonic aliases for a host or, in many respects, for its network interface(s). Later on, and with the DNS, it has become more useful to think to think of host names as an abstraction related to the system with, e.g., interfaces and their addresses as associated information. But things associated with those hosts were simply identified in terms of their locations: the user@host of an email address, a host name and file system reference to a file, and so on.

As originally conceived[1] , URLs were rather clear examples of the "identify by location" model. At this point, there is probably no point trying to reexamine whether that was ever a good idea. Given the rather poor state of generalized directory systems at the time and that the IETF had already discovered (via email addresses) the scaling issues associated with putting fine-grained identifiers[2] into the DNS, the decision is at least understandable. In today's Internet, it is important to understand that URLs are often treated, especially by some CDNs, as abstractions and remapped to find copies of "the same" object at different locations than the one specified by the host part (FQDN) of the authority component[3].

Because URLs also specify an explicit protocol or access method, recent discussions about using multiple transports with given URIs are another example of the consequences of the identification by scheme and location model – there is a requirement for inventing mechanisms to determine which protocols or schemes are "really" applicable Those decisions should arguably be under the control of the resource, not a matter of client-guessing (a lesson we learned with email routing more than 30 years ago[4]) but use of URLs as identifiers leaves little choice.

That view is probably slightly different from the one outlined in the first bullet of the solicitation, but they are probably ultimately the same topic. It is worth nothing that, while the solicitation refers to URIs, the idea of identifying a resource independent of its location or access method is exactly what URNs were supposed to be about, so the URL-URN distinction may be important.

More generally, the alternative of identifying the object and then allowing some database or process that is invisible to the ordinary user to control access, retrieval, or other actions is provided by a number of systems. Some that have general applicability and are familiar around the IETF include URNs[5], DOIs[6], the Digital Object Architecture more generally[7], and ARC[8] and WARC[9] , but there are several others with various degrees of adoption. Others, including ISBNs, ISSNs, and NBNs[10], are also familiar and have more specific applicability. It is not clear to me that URNs (or any of these other approaches) are the answers, but it would be unfortunate to start a new phase of the discussion without an understanding of work the IETF has already completed and seen deployed.

Whatever architectural choices between locators (even if abstracted) and more specific object identifiers the IETF makes going forward, there will be important questions about how changes should be deployed and transitioned to and whether that is even possible.

(2) The "just use the DNS because it is deployed and works" syndrome

During the last decade or two, decisions have been made many times to support particular types of identifiers or near-identifiers by using the DNS. Those decisions have often not involved careful considerations of DNS functionality and alternatives: the DNS is available and support is nearly ubiquitous, so why do anything else? The concern even applies to domain name use in URLs as DNS labels have evolved from naming network resources to being based on the names of products or associated with the content of particular web pages. Cumulatively, those decisions have led to expectations of the DNS and stresses on it, changes that have, in turn, sometimes resulted in suggestions for further DNS modifications. In what may approximate a positive feedback loop, some of those suggestions are impractical, at least without even further DNS changes. I observe that the Thursday session of the DNSOP WG at IETF 99 was devoted to proposals that appear to be part of this phenomenon. Parts of this issue are discussed at more length in a current Internet-Draft[11].

(3) Internationalization, natural language, and natural people

Perhaps more as the result of good luck than any particular planning or forethought, the subset of the Latin script contained in the ASCII repertoire, includes, with few exceptions, only characters that are easily distinguished from each other, even by people who do not use that script on a daily basis[12]. Allowing and encouraging use of non-ASCII characters on the Internet inherently involves us with scripts that are at least partially derived from others[13] or parts of scripts expanded from more basic elements[14] and that hence share character forms, small variations that are significant but that might be interpreted as type style variations by those familiar with other scripts[15]; scripts that are used differently

with different languages; scripts that require special consideration because of writing directionality or character joining conventions; and so on. These issues are fundamental to the writing systems involved and are typically many centuries old. Almost independent of the particular script or language, writing conventions and orthography evolve over time and often differently in different places. As an extreme example, some languages have been written in two or three different scripts during the last hundred or so years. Less extreme examples (even in Latin script) are illustrated by the divergence between British and American English and French as written in France as compared to that written in Quebec. There are a few additional problems consequent of Unicode decisions about character and string encoding (one particular set of those issues has been described as "non-decomposing characters"[16]), but they are far less significant than the variations that have little or nothing to do with coding or computer systems.

As we move from written expressions of language to spoken ones that are to be interpreted by machine, gestures, and so on, things because even more complex because correct usage may depend on context, with interpretative contexts. For example, should my spoken description be interpreted in the light of who I am (or my "normal" locale, language, and speech patterns or where I am. This is not a new problem: a different aspect of it has become obvious to almost everyone who has tried to use a search engine in a place far from home and had difficulty looking for something that should be interpreted in the home context or a third one.

Those relationships are rarely an issue in dealing with running text – sentences, paragraphs, or longer material -- in ordinary human languages. If nothing else, people are good at making inferences and adjusting for small and typically predictable variations. However, when identifiers (or other mnemonics) are involved, things become more complicated. Strings are often not long enough to establish context, there may be no language identification (and the identifier or mnemonic strings may not actually be words from which accurate language inference are possible), and systems to link the strings to other information may require exact matching rather than trying to support equivalence of strings that users know are the same (the DNS is a typical example of all of these problems).

However, an additional complication, even with identifiers, is that people will often see what they expect to see. A user who expects to see only letters from the ASCII repertoire and who has not developed a good case of paranoia is likely to see "paypaḷ", "g00g1e", or even "ซรก" as "paypal", "google" and "usa" respectively, ignoring the spare dot and other clues, the substitution of digits for letters, and a string from an entirely unrelated script as spots on the screen, author idiosyncrasies, or the use of an extremely fanciful type style[17].

The argument for staying with ASCII identifiers to make those identifiers clear, unambiguous, and not requiring extensive special treatment in handling can be carried even further, into a restriction to all-numeric identifiers (perhaps with separators among groups of digits). Even when those systems allow use with local digits, mappings among sets of digits are sufficiently precise that the local digits can be dealt with as a localization issue rather than as part of the identifier. It is not an accident that, e.g., the core part of DOIs and ISBNs and ITU telephone number and network identifiers are entirely numerical (except for a "/" separator in the DOIs) On the other hand, it may be worth remembering that mnemonics for ARPANET hosts were created because many people had trouble remembering the numeric forms and that the problematic numeric forms at the time only had a three decimal digit range.

(4) User identifiers and protocol identifiers

One conclusion from the discussion of identifiers that use characters other than a restricted subset of ASCII[18] is that one should carefully consider in each case when the non-ASCII characters are really necessary and avoid them otherwise. In particular, they should be avoided in keywords and identifiers that are used in protocols in ways that normally makes them invisible to end users. That principle was stated by the IAB over 20 years ago[19], however it is important to note that our understanding of it, its importance, and the tradeoffs involved has evolved significantly since then, and more generally since the IAB's statement at about the same time that "Designs should be fully international..."[20] without, in retrospect, even the vaguest of clues about what that would mean in practice. It may be time to revisit the question of what we should try to internationalize and where anything other than ASCII characters should be avoided, rather than making extremely ad hoc decisions for every protocol, whether that becomes a topic for this workshop or not.

It is perhaps notable in that respect that the PRECIS documents[21] do not appear to address the question of when it is more appropriate to just avoid non-ASCII characters rather than, e.g., selecting the appropriate profile. They instead assume that the choice to use non-ASCII characters is already made and a given. That is an entirely reasonable approach; what may be less reasonable is that we have given the community no clear guidance on that subject (other than for the DNS) in more than 20 years given how much we have learned, and the Internet has evolved, in that time.

(5) The mismatch between identifier-resolution systems (especially the DNS) and user expectations of non-ASCII identifiers

Many, if not most, of the challenges we have had with internationalized domain names (IDNs) since they were first deployed may be reflected in other identifiers as they come into wider use, attract those with an interest in identifying or utilizing possible exploits. With two exceptions, noted below, there is every reason to believe that issues we are seeing now (and have seen in recent years) with the DNS will eventually turn up with other protocols and applications that use non-ASCII textual identifiers.

Those users are sensible people. If an identifier looks like a word in their language, and the word can be acceptably spelled in three different ways, they expect any of the thrre that they happen to use to match the identifier. If European digits can be substituted in everyday life for Arabic or Chinese ones, then an identifier containing digits from one set (or maybe even a mixture of them) one should match the same string with the other set included. As far as I know, we have not yet encountered demands for the capability with the DNS, but more traditional synonyms, translations into other languages, transliterations into other scripts, and phonetic transcriptions should work too. Perhaps we could solve some of the problem by making a requirement that identifiers never be allowed to be words but there is ample reason to believe that such an idea would be a non-starter. , Unless we can develop a basis for saying "stop here" and having it be effective, demands of those types are probably just a matter of time.

If a language is written in one script in 2017, a different one in 1990 (and, in some locations, long after that), and another one in 1931, or if the forms of some (but not all or even a large percentage) characters were changed in the early 1950s, and users can remember words, terms, or writing styles from each of those periods, they may expect all of those to match – to be able to use one as input but have the lookup find another stored in the database – too. Perfectly sensible.

Unfortunately, the DNS (and most other systems that are implemented on computers that really are about identifiers) use exact match principles for which those sensible ideas do not work, at least reliably. In some cases, we can devise rules about what forms are likely to be present in the database and train users to adapt to them. That was exactly what we did when we told Unix users that, if a command or keyword was typed in upper case, it generally would not work. If is what the JET work[22] proposed for Chinese: a domain name could be in traditional characters or simplified characters, but no mixtures and, if one didn't know which one, the registration entity had to make special provisions that might not always work in a consistent and predictable way. Arguably, that hasn't worked out: those sensible humans who know perfectly well how their languages and writing systems work[23] keep mixing the simplified and traditional forms when they enter labels. Actual changes of script often work out better, especially if they are associated with social or political upheavals and the older script carries bad associations.

Especially when whatever conventions or protocol extensions can be invented for making labels match that are not bit-string identical end up working inconsistently with different protocols, or of a set of a half-dozen strings that a user expects will be treated as "the same", two are and the other four are not[24], the problem, in some respects, gets worse because the ones that do work reinforce the user view that they all should. Whether blocking (preventing use of) all but one is better or worse is an open question. One can easily imagine a user who actually knows, abstractly, what the identifier is trying one spelling, script, set of character forms, or other representation after another in the hope of finding a match, remembering that, in many circumstances, there can easily be dozens or hundreds of combinations, while expressing increasingly strong feelings that these are the sort of things computers are supposed to help people do, not make worse.

The problem will not go away. It is integral to the usability of systems in which identifiers are allowed based on a broad range of languages, scripts, and writing systems. In some scientific and information retrieval environments, the solution is to define a single (official, canonical) term that is used as the identifier and then create multilingual thesauri that can be used to find that term from any of the plausible variants. However, that approach, however realized, is one of a canonical identifier and a collection of alternate forms that can be used for it, not different forms of the identifier.

We can continue to ignore that underlying problem, but that will certainly not cause it to get better and may cause it to get worse.

I mentioned above that there were two issues unique (or far) to the DNS. One is that we established an ACE[25] form using the Punycode algorithm that has the effect of establishing a canonical form in most, but certainly not all, relevant cases, which most of the discussion above pointing to exceptions where it does no good. But it is almost certainly better than nothing. The PRECIS identifier profile does much the same thing for other types of identifiers. The second is that a "DNS marketplace" has evolved in which identifiers are assumed to have value independent of their identification function and often do, with significant amounts of money and numbers of vested interests involved. That situation is not as inherently bad as some of us believe at least some of the time, and this is not the right place to try to analyze the tradeoffs or what the alternatives might have been, but, when identifier strings are monetized, there are incentives for behavior, some of which might make the use of those strings as identifiers more difficult or less reliable, with regard to those strings.

(6) Naming authorities

I can identify one more topic that I think is nearing the critical path but that is probably less appropriate for a workshop than some in-the-hall brainstorming or discussion within the IAB and/or IESG. If identifiers are expected to be unique or to identify unique objects or resources (or families of them), some type of registration mechanism and an authority for it is almost always necessary. It is possible to design structured identifiers so that most of that administrative /authoritative function can be distributed, which is exactly what the DNS and, e.g., ISBN and ISSN, do, but there still has to be an entity responsible for the top-level identifiers or the mechanisms for allocating them.

The Internet's model for dealing with identifier registration has involved IANA since before there was an IETF. However that model was designed around the (mostly unwritten) principles that the identifiers themselves should be assigned by IANA, not subject to specific requirements from applicants, and, in particular, that the identifier strings themselves were not commodities. Disputes, if any, were about whether something should be registered and assigned an identifier, not about what that identifier should be. Those principles were motivated by, among other things, IANA's not wanting to be in the middle of disputes about who should "own" which names and to avoid even the suspicion of conflicts of interest involving applications and, e.g., IAB membership[26]. In more recent years, the process for approving registration requests has shifted to the IESG and its designees and, for many registries, the principle of IANA-assigned identifiers has eroded in the direction of registrant requests for particular identifiers and IANA assigning them unless there are identified conflicts or other problems.

It is not clear, despite three consensus versions of model registration approval rules[27], and very large numbers of discussions and intermediate drafts leading up to each one, that we have an adequate model, especially in two areas. First, when a registry must deal with subject matter about which there is not enough knowledge in the IETF community to make any consensus (IESG, IETF-wide, or otherwise) meaningful, it is not clear what we should do. If there are controversies, relying on a single expert seems very risky along a number of dimensions and violates our claimed prohibition on appointing "kings" even in narrow areas. While the appeal procedures could presumably be used, they may be too slow and heavyweight in practice and, more important, if there are substantive issues involved that are associated with topics very different from network technology, thee is no reason why the IESG and IAB would be competent to perform an evaluation. Second, it is not clear that existing arrangements are adequate to protect designated reviewers, the IESG, and the IETF community against conflicts of interest or from claims that a decision was motivated by self-interest by various parties. So, while this topic is the furthest afield from the IAB's solicitation announcement, it is probably nonetheless part of the picture that must be assembled if we are to move forward in the identifier area.

A Final Observation

While I don't think it is a universal solution, I believe that we could make great progress on several of these issues, including some mentioned in the solicitation but not explicitly addressed above, by trying to be a lot more clear about the difference between identifiers and mechanisms used to find and choose the desired identifier and then by separating the two. Especially in areas where a good deal of context or language or character mapping or other efforts are required to find the identifier (or the object) the user is really seeking, mechanisms that assist the user in identifier-finding, rather than expecting the user to work with the identifiers directly may turn out to be the only viable approaches. Fortunately,

whether one sees them as having been applied in this way or not, the community has a great deal of experience with such mechanisms. Users rarely use URLs or domain names but instead rely on search engines (and rarely know the difference). While they have been eclipsed in recent years by search engines, the community has considerable experience with directory services, various types of indexes, and so on. Perhaps it is time to ask slightly different questions than ones that seem equivalent to “how do we make identifiers work better in environments that are increasingly hostile to them”.

1 See RFC 1738 and earlier documents

2 In that case, user names.

3 RFC 3986

4 See discussion in RFC 974 (January 1986)

5 RFC 8141 and elsewhere

6 See RFC 7669, at least one expired I-D, and other work.

7 See https://www.internetsociety.org/doc/overview-digital-object-architecture-doa and draft-durand-doa-over-dns. The latter is a bit strange in this context because a key element of the DOA is its own resolution system, but precedent for ignoring that and using the DNS (and even the web) can certainly be found in the very popular http://doi.org/… lookup mechanism.

8 See http://archive.org/web/researcher/ArcFileFormat.php

9 Draft-kunze-warc-00 (expired)

10 RFC 3188

11 draft-klensin-dns-function-considerations

12 If Western European writing systems, notably English, has stuck with the characters used during the late Roman Republic, there were be even fewer issues.

13 The relationship among Greek, Latin, and Cyrillic scripts is the one most often cited, but there are many other examples. Indeed, most scholars of the development of writing have concluded that all "alphabetic" scripts, including both those that explicited represent vowel sounds and those that don't, have a common origin if one goes back far enough.

14 Compare what we describe as "Latin script" today and the characters that would have been recognized by .e.,g Cicero as part of his writing system.. There are many other examples with other scripts, particularly those alphabetic or phonetic ones that started out being used for one language and then were extended to be used for other languages with different phonemes.

15 As an example, are assorted dots, attached and detached rings, wiggles, and other "decorations" significant or merely decorate? The general answer is "sometimes, with knowledge of the script, and sometimes the language and writing system, needed to be confident about a more specific answer.

16 draft-klensin-idna-5892upd-unicode70. That draft is now outdated (not merely expired); I will post a new version before the workshop if that appears to be relevant. While its specific recommendations still appear to be reasonable, the IAB statement on the subject (https://www.iab.org/documents/correspondence-reports-documents/2015-2/iab-statement-on-identifiers-and-unicode-7-0-0) is even more outdated because a number of important cases, and sets of case, were not yet identified when it was written.

17 For all the time and energy that has gone into discussions of confusion among characters forms in the last decade, the reality is that what is or is not likely to be confused depends significantly on the context in which characters appear, not just individual pairs of characters or code points, on such issues as choices of type styles and sizes, and on the amount of knowledge individual users possess and how carefully they apply it. In general, none of those factors are under the control of those who define identifiers or rule for registries and what is permitted.

18 Ig may be important to remember that the so-called "preferred syntax" of the DNS (and the host naming rules that preceded it) do not allow ASCII and exclude everything else. Instead, they allow only ASCII letters, digits, and the hyphen, restrict where digits and the hyphen can be used, and, with the exception of the period used as a label separator, exclude akk ASCII control characters, punctuation, and other symbols.

19 RFC 2130

20 RFC 1958

21 RFC 6885, draft-ietf-precis-7564bis-10

22 RFC 3743 and then RFC 4713

23 Actually, experience (including a few episodes that have shown up during IETF work) indicates that they often don't and that ability to read and write a language is often only loosely correlated with understanding the details of how the writing system works, but their behavior is conditioned on what they think they know and what intuitions they have.

24 Because of the combinatorial explosions that occur when a label at one label of the hierarchy that may have several different form a user would consider equivalent is combined with several other labels with similar properties to form an FQDN,, those numbers are far too small. In practdice, one could easily have a few DQDNs that would work as the user would predict and hundreds that would not. Or one could try putting all of them into the DNS and end up with a horrible management problem.

25 ASCII-Compatible Encoding

26 The case that has most often been discussed in recent years, the use of ISO 3166-1 to determine ccTLD names, is actually just a corollary of those principles.

27 RFCs 2434, 5226, 8126