SAC1~~XX~~23
SSAC Report on the Evolution of Internet Name Resolution

A Report from the ICANN Security and Stability Advisory Committee (SSAC)
~~DD~~15 ~~Month~~December 2023

Preface

This is a report to the ICANN Board, the ICANN organization staff, the ICANN community, and, more broadly, the Internet community from the ICANN Security and Stability Advisory Committee (SSAC) about the evolution of Internet name resolution.

The SSAC focuses on matters relating to the security and integrity of the Internet's naming and address allocation systems. This includes operational matters (e.g., pertaining to the correct and reliable operation of the root zone publication system), administrative matters (e.g., pertaining to address allocation and Internet number assignment), and registration matters (e.g., pertaining to registry and registrar services). SSAC engages in ongoing threat assessment and risk analysis of the Internet naming and address allocation services to assess where the principal threats to stability and security lie, and advises the ICANN community accordingly. The SSAC has no authority to regulate, enforce, or adjudicate. Those functions belong to other parties, and the advice offered here should be evaluated on its merits.

Table of Contents

Executive Summary

New technologies are changing how name resolution happens on the Internet. The DNS remains the prominent, or default, naming system for the Internet, but alternative naming systems are in use as well. This is nothing particularly new, as there have always been naming systems besides the DNS in use throughout the Internet's history. These alternative naming systems use the same syntax as the DNS, dot-separated labels. There are many motivations for copying this syntax, but the primary reason is because designers of these alternative naming systems wish to benefit from the existence of software applications built to receive DNS names as input.

This has the potential to create situations where the same name exists in DNS and in an alternative system, potentially causing name collisions. However, there is only one domain namespace and its referential integrity is important for Internet users and for the stability and security of Internet names. Thus, as alternative naming systems increase in popularity their use threatens to increase ambiguity in the shared single domain namespace. This increased ambiguity in Internet naming threatens to undermine the trust that users have in Internet identifiers and the services that rely on them.

Additionally, names are becoming less visible to Internet end users, yet they remain vital to the security and stability of Internet infrastructure. Technologies such as QR codes and URL shorteners offer great utility to Internet users while also obscuring the underlying domain names used and creating new opportunities for malicious behavior. Meanwhile, QR codes and URL shorteners use domain names to access the Internet resource, even if the human user does not see it.

These are the two main trends that the SSAC identifies in this report. The same name can resolve in different ways (ambiguous name resolution), and names of service endpoints are less visible (names are less conspicuous to end users). It is the combination of these two trends that fundamentally threatens to undermine confidence in services on the Internet.

The SSAC then identifies proposals being considered to help facilitate domain namespace coordination, and applauds ICANN in the activities they have engaged in to facilitate namespace coordination. Finally, the SSAC recommends that ICANN continue to track these topics, and continue to encourage collaboration and coordination among the various namespace communities.

1 Introduction¶

‘When I use a word,’ Humpty Dumpty said in rather a scornful tone, ‘it means just what I choose it to mean--neither more nor less.’¶

‘The question is,’ said Alice, ‘whether you CAN make words mean so many different things.’
–Lewis Carroll, ~~Alice's Adventures in Wonderland~~ Alice in Wonderland~~Through the Looking-G;ass and What Alice Found There~~ Through the Looking-Glass and What Alice Found There

In the traditional model of DNS resolution, a DNS library is included in operating systems and all applications on that host use this library to resolve names. While the library’s operational parameters are sometimes configured by the end user, they are more often configured through the use of the Dynamic Host Configuration Protocol (DHCP). Applications themselves do not perform DNS resolution, but instead use the operating system’s library. As the SSAC discussed in *SAC109: The Implications of DNS over HTTPS and DNS over TLS* (SAC109), this traditional model of DNS resolution is changing. This report expands on SAC109 by exploring naming systems other than the global DNS, what we refer to as alternative naming systems.

In the traditional model of DNS resolution, users have little control over their DNS settings. Some technically literate users may choose settings that differ from the defaults, but there has been little incentive to do so, and the vast majority of users have their DNS settings configured for them by administrators via a protocol such as DHCP.

Many alternative naming systems in use today come bundled with the specific applications that use them: a particular alternative naming system is often tied to a corresponding application, and this application often bypasses administrator-controlled settings and any pre-configured DNS settings. For example, the Tor Project uses its own naming system that bypasses traditional DNS resolution. A user can install the Tor Browser, and it will use the Tor naming system for names ending in .ONION, while forwarding any other names to the local DNS library. The application developer makes the choice of which naming system to use without the user even knowing that they are using an alternative naming system nor understanding potential implications.

This report explores the effects and implications of these alternative naming systems. Much of the SSAC’s concern with alternative naming systems lies in the ambiguity that inherently comes with the increased use of alternative name systems that use the domain namespace. The ways in which a computer performs name resolution are not as clear as they once were, in large part because the roles of the operating-system-provided DNS libraries and applications have evolved. Name resolution within computers can now follow different paths and the resolution path chosen can sometimes determine the service that is actually reached. This has important implications for users, particularly in the critical areas of predictability and security.

Names play an important role in how users trust the services they use on the Internet. However, names are becoming less visible, or at least less conspicuous, to users. Many of the interfaces where users previously saw domain names now hide domain names from users.

Thus, the SSAC recognizes three important trends occurring at the same time in Internet naming: domain name resolution is becoming more ambiguous, domain names are important for trust and security, and users do not encounter domain names in applications as much as they once did.

The Internet is evolving and naming will evolve with it. While the global DNS is currently the default Internet naming system, it may not remain the default forever. There are legitimate reasons to experiment with new naming technologies that push against the status quo. This document explores the many motivations to develop new naming systems as well as why most of these new alternative naming systems adopt the same syntax for names as the DNS. The DNS will continue to change and evolve, and it is likely that some future default naming system for the Internet will incorporate some of the elements found in current experiments.

1.1 Past SSAC Publications on the Domain Namespace

This is not the first publication that the SSAC has written on the topic of namespace ambiguity. This section reviews and briefly summarizes past SSAC publications on this topic.

1.1.1 SAC062: SSAC Advisory Concerning the Mitigation of Name Collision Risk

SAC062 discusses the risk of name collisions that can arise through the delegation of new top-level domains (TLDs). It discusses the lack of coordination with other groups, high-risk strings such as .HOME and .CORP, and how ICANN might mitigate the risk of new TLD delegations through trial delegations.

SAC062 contains three recommendations.

Recommendation 1: ICANN should work with the wider Internet community, including at least the IAB and the IETF, to identify (1) what strings are appropriate to reserve for private namespace use and (2) what type of private namespace use is appropriate (i.e., at the TLD level only or at any additional lower level).

Recommendation 2: ICANN should explicitly consider the following questions regarding trial delegation and clearly articulate what choices have been made and why as part of its decision as to whether or not to delegate any TLD on a trial basis.

- Purpose of the trial: What type of trial is to be conducted? What data are to be collected?
- Operation of the trial: Should ICANN (or a designated agent) operate the trial or should the applicant operate it?
- Emergency Rollback: What are the emergency rollback decision and execution procedures for any delegation in the root, and have the root zone partners exercised these capabilities?

- Termination of the trial: What are the criteria for terminating the trial (both normal and emergency criteria)? What is to be done with the data collected? Who makes the decision on what the next step in the delegation process is?

Recommendation 3: ICANN should explicitly consider under what circumstances un-delegation of a TLD is the appropriate mitigation for a security or stability issue. In the case where a TLD has an established namespace, ICANN should clearly identify why the risk and harm of the TLD remaining in the root zone is greater than the risk and harm of removing a viable and in-use namespace from the DNS. Finally, ICANN should work in consultation with the community, in particular the root zone management partners, to create additional processes or update existing processes to accommodate the potential need for rapid reversal of the delegation of a TLD.

All three recommendations from SAC062 are now closed. Recommendation 1 was resolved after ICANN's Office of the Chief Technology Officer (OCTO) worked in the IETF DNS Operations Working Group to update RFC 6761, but were ultimately unable to gain traction. The recommendation was then closed as the community was not able to achieve consensus on how to move forward.[1] Recommendations 2 and 3 were closed after being considered by ICANN while developing the Name Collision Occurrence Management Framework.[2,3] The Name Collision Occurrence Management Framework was adopted by the ICANN Board's New Generic Top-Level Domain (gTLD) Program Committee (NGCP) in July of 2014.[4]

1.1.2 SAC090: SSAC Advisory on the Stability of the Domain Namespace

SAC090 is the most significant SSAC publication on namespace ambiguity. It contains a short history of domain names on the Internet, including their usage prior to the DNS, and discusses the main issues that arise from domain namespace ambiguity. It contains two findings and four recommendations summarized below.

Finding 1: The SSAC finds that uncoordinated use of the domain namespace in overlapping environments can lead to ambiguity when those environments overlap and their names collide. This ambiguity threatens the stability of the domain namespace when processing agents cannot reliably determine "what to do" when presented with an identifier that is a syntactically valid domain name.

Finding 2: More specifically, the SSAC finds that the lack of adequate coordination among the activities of several different groups contributes to the domain namespace instability identified in Finding 1. [..]

---

[1] See RFC 6761: Special-Use Domain Names
[2] See Letter from Matt Larson to Rod Rasmussen, 13 January 2021, https://www.icann.org/en/system/files/correspondence/larson-to-rasmussen-13jan21-en.pdf
[3] See Name Collision Occurrence Management Framework, https://www.icann.org/en/system/files/files/name-collision-framework-30jul14-en.pdf
[4] See ICANN Board Approved Resolutions | Meeting of the New gTLD Program Committee | 30 July 2014, https://www.icann.org/en/board-activities-and-meetings/materials/approved-resolutions-meeting-of-the-new-gtld-program-committee-30-07-2014-en

Recommendation 1: The SSAC recommends that the ICANN Board of Directors take appropriate steps to establish definitive and unambiguous criteria for determining whether or not a syntactically valid domain name label could be a top-level domain name in the global DNS.

Recommendation 2 defines the scope of work for recommendation 1.

Recommendation 3: Pursuant to its finding that lack of adequate coordination among the activities of different groups contributes to domain namespace instability, the SSAC recommends that the ICANN Board of Directors establish effective means of collaboration on these issues with relevant groups outside of ICANN, including the IETF.

Recommendation 4: The SSAC recommends that ICANN complete this work before making any decision to add new TLD names to the global DNS.

At the time this document was published all four recommendations were waiting to be closed pending the work being carried out in the Name Collision Analysis Project (NCAP).

1.1.3 SAC113: SSAC Advisory on Private-Use TLDs

SAC113 discusses private-use TLDs and the issues that can arise from their use. It contains a single recommendation to the ICANN Board to reserve a string from delegation in the DNS root zone in perpetuity.

Recommendation 1: The SSAC recommends that the ICANN Board ensure a string is identified using the criteria specified in Section 4.1 and reserved at the top level for private use. This particular string must never be delegated.

The SSAC proposes the following criteria for the selection of the string:

1. It is a valid DNS label.
2. It is not already delegated in the root zone.
3. It is not confusingly similar to another TLD in existence.
4. It is relatively short, memorable, and meaningful.

At the time this document was published, ICANN org was in the process of implementing this recommendation, but a string has not yet been chosen for this purpose.¶

2 Definitions

The terminology used in this document is consistent with the terminology in RFC 8499. Specifically, the definitions of the terms **domain name**, **global DNS**, **label**, **naming system**, and **private DNS** are taken from RFC 8499. Additional terms not defined in RFC 8499 are defined below for their use in this document.

name resolution - The process of translating a name to a network resource such as an address.

namespace - A set of names used to identify and refer to objects.

domain namespace - The set of all possible names that could be assembled using a common syntax defined for domain names.[5]

DNS name - A domain name that could exist in the global DNS.

alternative naming system - Any naming system other than the global DNS used by computers to resolve a name. For example, an alternative naming system can be used to map a name to a network service.

name provisioning - The process of adding names to a naming system. For example, in the DNS this is the process by which names get added to zone files. For TLD zone files this process begins with a registrant registering a name at a registrar and ends with that name being added to the TLD zone file.

resolution context - The *view* of the domain namespace from a given application that is resolving names. The default resolution context used by most Internet applications is the DNS. For these applications if a name exists in both the DNS and an alternative naming system the name will resolve via the DNS. However, if an application uses a different resolution context it may perform name resolution using a different process and therefore receive a different answer.

2.1 Design Trade-Offs

Like all designers, designers of alternative naming systems make choices depending on their requirements and what they are hoping to achieve with their design. In essence, designers never get everything they want and trade-offs must be made. Additionally, designers do not always recognise they are trading off one aspect of a design for another. Recognition of the two trilemmas discussed in this section is not necessary for them to apply. To the extent they are representative of actual trade-offs, they are always active. However, these trilemmas are not laws of nature, and reasonable people will disagree as to their efficacy. They are included here for illustrative and pedagogical purposes.

In project management and software engineering a common trilemma used to illustrate trade-offs is the *Good, Cheap, Fast Trilemma*. Often depicted as a triangle, this trilemma illustrates the trade-offs that project managers make when executing projects. *Good* refers to the overall quality of the desired outcome, *Cheap* refers to the cost to achieve the desired outcome, and *Fast* refers to how quickly the desired outcome can be reached. No project can achieve all three of them.

[5] From SAC090: SSAC Advisory on the Stability of the Domain Namespace, "In formal graph-theoretic terms, the domain namespace constitutes a labelled directed rooted tree in which the syntax of the label associated with each vertex other than the unlabeled root is defined by RFCs 1035, 1123, and 2181. The term "nth level domain name label" refers to a member of the set of all vertices for which the path to the root contains n edges. For n=1 the term most often used is "top-level domain name label" or simply "top-level domain" (TLD). In this formulation, the term "domain namespace" refers to the complete graph consisting of all possible vertices and edges, not just those with which a specific use has been associated."

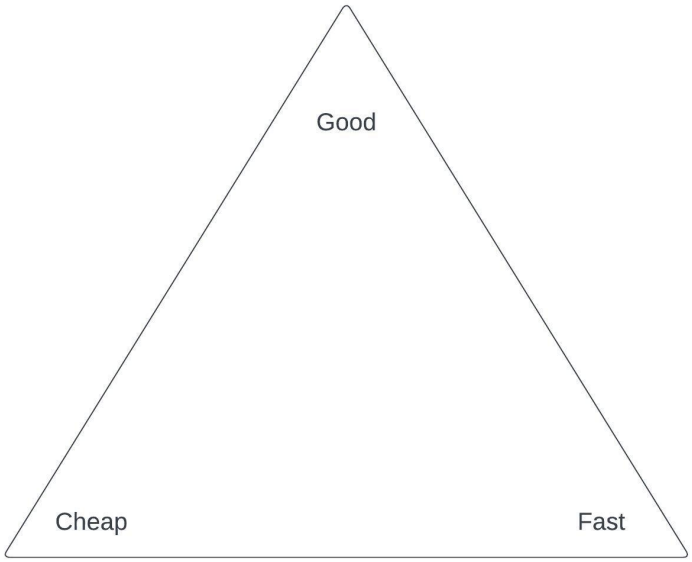

Figure 1: Good, Cheap, Fast Trilemma

¶

Zooko's Triangle is a trilemma first proposed by Zooko Wilcox-O'Hearn in 2001.[6] It can be a useful tool for understanding trade offs in designing naming systems. It postulates that the three fundamental properties desired in a naming system cannot all be achieved in the same naming system. According to Zooko's Triangle, trade-offs must be made when designing a naming system and any design can only achieve two of the three properties. The three properties are *Distributed*, *Secure*, and *Human Memorable*. *Distributed* means that the name does not rely on a single point of trust (i.e., a single root). *Human Memorable* means the names are meaningful and memorable to users. *Secure* means that a user, or their agent, can authenticate the name.

In terms of Zooko's Triangle, the global DNS in combination with DNSSEC is ideally Secure and Human Memorable, but it is not Distributed. There is a single DNS root zone and all trust in the global DNS is tied to it.

[6] See Internet Archive: Names: Distributed, Secure, Human-Readable: Choose Two, https://web.archive.org/web/20011020191610/http://zooko.com/distnames.html

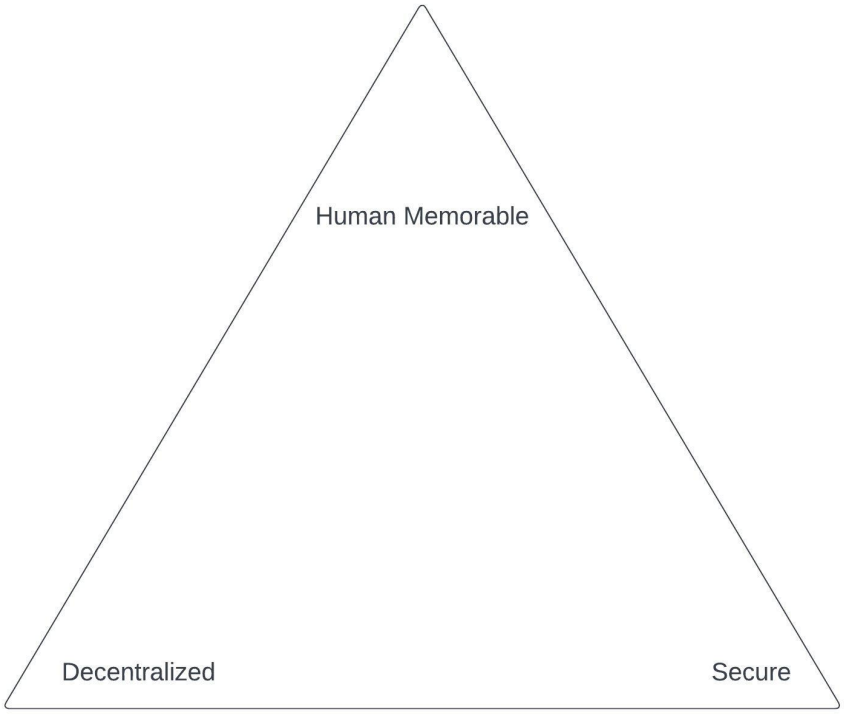

Figure 2: Zooko's Triangle

3 Motivations to Use Domain Name Syntax

The default name resolution context for most Internet applications is the global DNS. With some exceptions, these applications only perform resolution using the DNS. This mainstream use of DNS as the common resolution context helps ensure that domain names can be used as identifiers outside of computer interfaces. It also motivates designers of new naming systems to reuse the syntax of DNS names.

Advertisers rely on the mainstream acceptance of the global DNS as the default resolution context for domain names to connect customers to products and services. Domain names can be written down, memorized, and sometimes spoken out loud. In short, domain names are effective identifiers because there is a mainstream default resolution context. Designers of new naming systems want their names to be compatible with software and applications that already use DNS names.

Most people believe they are able to distinguish between a phone number and a domain name. However, because of the rise of smartphones and other devices that offer converged access to these networks even this has become more ill-defined. As more alternative naming systems get deployed that use the same syntax as domain names the distinctions become even less clear.

Table 1 below displays example strings and whether or not each string can be considered a domain name and/or a DNS name. Reasonable experts may disagree on the strict categorization of the strings shown in *Table 1* and this is partly the point. It highlights the difficulty of categorizing strings into these identifier categories without additional context, and the difficulty designers of applications have in determining how to handle different strings. In *Table 1* these strings are presented devoid of context. There is no additional information presented with the string to assist in determining how the string should be interpreted by a human or and handled by software. In practice, identifier strings are rarely presented without any additional context. Web browsers, for example, use URIs, where the URI scheme provides the context ("http" vs "mailto" vs "tel").

In *Table 1* the definition of *domain name* is taken from RFC 8499 and the definition of *DNS name* is taken from Section 2 of this document.

String	**Domain Name**	**DNS Name**	**Comment**
example.com	✓	✓	
example.com.	✓	✓	Trailing dot.
example.zip	✓	✓	File extension or TLD?
212.555.1212	✓	✓	Could be a telephone number
+1.212.555.1212	✘	✘	Does not follow LDH rule.[7]
192.0.2.1	✓	✓	Mimics IP address.
printer.local	✓	✘	RFC 6762
_printer.example.com	✓	✓	RFC 952 RFC 8552[8]
3g2upl4pq6kufc4m.onion	✓	✘	
example.삼성	✘	✘	Korean IDN for "samsung".
example.xn--cg4bki	✓	✓	A-labels for the above IDN.

[7] RFC 952 limits hostnames to the now well-known letter-digit-hyphen (LDH) set of characters. However, RFC 8552 refers to names beginning with underscores as 'reserved node names'.
[8] *Id.*

will.i.am	✓	✓	A singer.

Table 1: Categorization of Example Strings

4 Motivations to Change Internet Name Resolution

When the domain name system was initially developed and deployed in the early 1980s, there were technological constraints that strongly influenced its design. The names that were in use prior to the DNS were organized differently.[9] The DNS was designed hierarchically primarily so that it could support a large number of names. There were memory and processing constraints in the 1980s that limited how many names a machine could keep in memory at one time, and this limitation was behind this view of the merits of a structured hierarchy as compared to an unstructured flat namespace.

Its hierarchical distributed design facilitated one of the most important aspects of the global DNS: delegated governance. An operator could delegate the authority to govern an entire sub-tree by indicating different nameservers for a label within a domain. This hierarchical delegation resulted in structuring the governance of the DNS and is what allows delegated domains to have different policy regimes from other delegated domains and from their 'parent' domain.

Another implication of the adoption of a hierarchically organized nameserver structure was an iterative name resolution process. The first part of name resolution was a discovery process to identify the nameserver that serves the zone that contains the name in question, and the second part was to query that nameserver with the query name. The protocol design used the same query and response framework to both discover the authoritative nameserver and then query this nameserver for the desired response.

Not only have the original limitations behind the DNS design changed since its inception, but the requirements and desired behavior of the DNS have changed as well. There is increasing pressure to provide content to users at higher speed. This motivates changes to the DNS that promote faster resolution of names and a reduction in the number of queries that need to be resolved in order to serve content to the user.

In recent years there has been increased awareness of privacy considerations for users and the risk of information leakage via DNS queries. This has motivated changes to the DNS resolution protocol and how it is used.[10,11,12] Accordingly, these modifications have made many DNS transactions less susceptible to opportunistic eavesdropping.

[9] See Pre DNS naming, Presentation given by Roy Arends at ICANN 54 ccNSO Tech Day, https://meetings.icann.org/en/dublin54/schedule/mon-tech/presentation-early-naming-19oct15-en
[10] See SAC109: The Implications of DNS over HTTPS and DNS over TLS
[11] See RFC 7816: DNS Query Name Minimisation to Improve Privacy
[12] See RFC 9250: DNS over Dedicated QUIC Connections

We have also seen various efforts to manipulate the user experience by deliberately tampering with DNS resolution outcomes. This has led to a desire to make DNS name resolution less susceptible to third party manipulation[13] and has resulted in efforts to improve the ability to validate the answers provided by DNS resolution transactions.

Thus, whereas the original motivation for the design of the DNS namespace and name resolution protocol was to ensure that the DNS was viable for the technical capabilities of the time, we now see a number of new evolutionary pressures based on improved technical capability of the infrastructure: a desire to improve both speed and privacy of name resolution, and a desire to better verify the authenticity of DNS responses.

The DNS resolution process can be likened to a conventional database lookup, in that the query term is used to find an exact match across the collection of lookup keys, and if a match is found the corresponding data entry is returned. In such a model the response is constant. The wildcard construct in a DNS zone essentially implements a 'default' response which is used in the absence of an exact match, where the wildcard response remains constant. There has been some experimentation in altering this assumption of response constancy by configuring the server to generate a response based on client parameters (such as IP address) or the query label itself. In such a model of dynamically-generated DNS responses, the DNS resolution protocol has a strong resemblance to a form of remote procedure call, where the query triggers the DNS server to execute some actions based on the query label. This has been used to reduce the time for network transactions by combining DNS name resolution and a simple query/response transaction into a single DNS resolution operation. Other experiments have extended this model by placing state into the DNS server, allowing the DNS resolution protocol to be used as a medium for traffic tunneling or remote execution, bypassing existing network-level filters.

Governance of the DNS is another aspect where some experimenters want to see changes. The current structures that govern the DNS and the namespace have enfranchised some entities as registry and registrar operators, while excluding others. There has been exploration into mechanisms that use different name governance structures and that would place new or different entities in governance roles. For example, some distributed web platforms have embedded the principle of decentralization into their governance structures by creating “Decentralized Autonomous Organizations” (DAOs) to manage decision-making via member voting as a way of deliberately departing from centralized authority models.[14] There has also been research into cryptographically ensuring orderly transfers of power using blockchains.[15]

---

[13] See RFC 9364: DNS Security Extensions (DNSSEC)

[14] See Faqir-Rhazoui, Youssef, Javier Arroyo, and Samer Hassan. “A Comparative Analysis of the Platforms for Decentralized Autonomous Organizations in the Ethereum Blockchain.” Journal of Internet Services and Applications 12, no. 1 (October 1, 2021): 9. https://doi.org/10.1186/s13174-021-00139-6.

[15] See Campanelli, Matteo, Bernardo David, Hamidreza Khoshakhlagh, Anders Konring, and Jesper Buus Nielsen. “Encryption to the Future.” In Advances in Cryptology – ASIACRYPT 2022, edited by Shweta Agrawal and Dongdai Lin, 151–80. Lecture Notes in Computer Science. Cham: Springer Nature Switzerland, 2022. https://doi.org/10.1007/978-3-031-22969-5_6.

These and other initiatives can be motivated from a desire to prevent perceived concentrations of power; one blockchain organization has warned that, “Internet participants will bump up against the limits of today's highly centralized trust models, where major web services are administered primarily by large, monopolistic for-profit corporations.”[16]

The root zone of the DNS also suffers from a lack of diversity in allocation mechanisms. The DNS has primarily two allocation mechanisms. Generic TLDs are allocated based on policies and procedures established by consensus of those who seek to be the operator of the TLD. Country code TLDs are allocated to recognized ~~countries~~entities based on IANA policy.[17]

For those parties that cannot use either of these allocation mechanisms their only choice is to experiment with alternative naming systems that also provide alternative allocation mechanisms. There is no small set of TLD allocation mechanisms that will work for every party that wants a TLD.

Censorship resistance is another motivation of some experimenters. Websites and domain names can be taken down by any of several parties: the registrar, the registry, the nameserver operator, or the hosting company. This has led to technologies that promise different levels of permanence in naming and that can make the promise that once a name has been added to the system it cannot be removed. This, in turn, has motivated different kinds of distributed ledgers, such as blockchains, to be developed and deployed that can provide different assurances of permanence.

While such evolutionary pressures exist in the DNS environment, that does not mean that any such pressure will result in changes to the DNS or its resolution protocol. The global DNS is the default naming resolution system for the Internet, and there is a considerable operational infrastructure built around its administration and use. Thus, there are also pressures to resist change to the DNS and maintain the status quo in Internet naming. This resistance to change is generally commensurate with the size of the installed base and the scope and complexity of the change being contemplated.

This stasis of the installed base does not imply that all forms of evolutionary change in the DNS will be resisted. If the pressures for evolution can be addressed to some extent through small-scale changes that can be deployed incrementally, there is a greater likelihood that the change will be at least partially adopted. We have seen this with encrypted transports, such as DNS over TLS (DoT), DNS over HTTPS (DoH), and DNS over QUIC (DoQ), which are now used to improve the privacy of DNS resolution. Larger-scale changes that cannot be adopted in an incremental fashion, such as changes that define a new namespace, pose a harder challenge for adoption.

Innovation requires differentiation: any alternative naming system and its alternative associated name resolution protocol must be different from the DNS in a useful way. For an alternative

[16] See A Foundation for the Third Internet Era, https://www.decentralizedinter.net/blog/a-foundation-for-the-third-internet-era

[17] See Delegating or transferring a country-code top-level domain (ccTLD), https://www.iana.org/help/cctld-delegation

system to gain wide acceptance, it needs to stand out in some fashion and provide functionality or overcome some technical limitation of the DNS.

Innovation causes disruption: it may not be the intention of the promoters of an alternative naming system to disrupt the DNS, but any shift away from a clear default naming system for the Internet will inevitably cause some disruption, even if only because a new distinction exists that previously did not. This causes tension between those wanting to advance the state of the art and those wanting to minimize changes for the sake of stability. Such parties will naturally want the status quo to be maintained, and view any new entrants in Internet naming as a threat not only to the status quo, but also to the stability and resiliency of the domain namespace. These tensions are inevitable and successful innovation requires small evolutionary steps that maintain backward compatibility.

Not all movement is forward and not all change is progress.[18] What some people view as innovation and new technology others may view as meddlesome disruption with no benefit. It is impossible to know the future of the DNS, or of Internet naming more generally. Motivations to develop new and distinct alternative naming systems will continue to arise. The necessities of Internet users today will be different in the future, and these will drive development of new alternative naming systems. Likewise some of the alternative naming systems that are in use today, or that seem popular today, may no longer exist at some point in the future.

4.1 Crossing the Implicit-Explicit Boundary

Rarely are Internet identifiers presented with no accompanying context at all. For example, when end users sit down at a table in a restaurant and encounter a QR code instead of a menu they will likely understand why that QR code has been placed there and what it is expected to take them to. Likewise when a developer is writing an application to handle input from users there will be both limitations on the kind of input permitted, as well as a possible shared understanding of the format for that input. Telephone numbers may have different presentation formats depending on what country the number is in and what kind of number it is (e.g., mobile or landline). Developers can rarely expect users to input E.164 formatted telephone numbers when prompted, and instead have to accept different formats depending on the expectations of their users.

URI schemes can also be used to denote some limited context for a domain name. Beginning a URI with https:// tells user agents to contact the specified host using HTTP over TLS to form a connection, while a mailto: URI signals resolution of an MX record.[19]

When the global DNS was not only the clear default resolution protocol, but also the only one that almost all users and developers knew about, Internet naming application designers could safely assume that the DNS was the correct resolution protocol for names. However, as the Internet grew and became the dominant form of communication, and as more alternative naming

[18] A slightly less personal and more economic tone than the Ellen Glasgow quote, “All change is not growth, as all movement is not forward.”
[19] See IANA Registry of Uniform Resource Identifier (URI) Schemes, https://www.iana.org/assignments/uri-schemes/uri-schemes.xhtml

systems were introduced, these implicit assumptions about context that had been fixed became variable. Designers had to think about which resolution protocol to use, but the DNS had no way to represent these new explicit requirements to provide this necessary context.

There is no way a name by itself can include context that can signal which resolution protocol should be used. Any context that could help an application determine the intended resolution protocol of a name must be implied from the environment in which the name appears.

This generic and recognised problem in computer science literature has been referred to as "Crossing the Implicit-Explicit Boundary."

> Suppose you were to develop a record system for all graduate students in your university. For each student you assign a record that includes name, social security number, year, and department. At first everything works swimmingly well. Your program becomes famous – so much so that all the other schools in your neighborhood decide to use it. So far there is no problem. But now suppose a motion is introduced to allow students to cross-register among schools, requiring all the various copies of the program to be able to communicate. It is clear that to support this expanded use, another entry will be needed in the database – a new field, per student-class-registration, to identify the school at which they are registered.
>
> This example illustrates a situation that is tremendously widespread in real-world practice: of needing to add a field (slot, variable, whatever) to a representational data structure so as to make explicit something that was previously implicit, so as in turn to allow the record or system to be used in settings in which a property that was originally and tacitly assumed to be fixed, can now be changed to one in which it is explicitly recognized to be variable. And one can never know all the slots one might need in advance.[20]

There is no way to add a field (slot, variable, whatever) to a domain name *in situ* that could signal to an application *which* resolution protocol is intended. Thus, designers of alternative naming systems have resorted to signaling intended resolution protocol through the use of specific TLDs (e.g., .ALT, .ONION, .LOCAL). The IETF could add another field to the DNS protocol, but that would still be using the DNS protocol. There is nothing above all naming systems, no generalized Internet naming signaling mechanism that could provide this functionality for all naming systems in use on the Internet.¶

5 Examples of Alternative Naming Systems

This section contains examples of alternative naming systems. The examples chosen here are not exhaustive. Instead they were chosen to highlight specific characteristics of alternative naming systems the SSAC believed were important to illustrate. Unless otherwise noted, none of the

[20] See Smith, Brian Cantwell. On the Origin of Objects. First Edition. Cambridge London London: The MIT Press, 1996, page 43.

top-level domains listed in this section are in the DNS root zone. For a more detailed technical investigation into various decentralized naming systems, including the Ethereum Name Service and the GNU Name System, see the ICANN OCTO commissioned research.[21]

5.1 Multicast DNS

Multicast DNS (mDNS) is a name resolution mechanism for local-area networks (LANs).[22] It uses the .LOCAL top-level domain, which is included in the IANA Special-Use Domain Names registry. It is intended to help LAN users resolve names for devices connected to their LAN, such as printers, so it requires little to no administration or configuration on the part of users or device makers. Despite its name, mDNS does not use the DNS protocol for resolution nor is it designed to interoperate with the global DNS. It is intended to be used only in a local context.

mDNS is in wide use with different implementations made by different software and hardware vendors. Apple uses mDNS for their Bonjour protocol, and many devices intended for use in home and office environments use mDNS as their sole network discovery mechanism.[23]

mDNS provides local naming for devices without requiring a nameserver. Unlike DNS, there are no distinctions between nameservers and resolvers, and because it is only meant to be used on LANs, it can use local IP multicast. The same name can exist on multiple LANs without problem, which makes it ideal for LANs and unsuitable for the Internet. mDNS is an example of scoping in the DNS resolution environment through the use of a distinguished domain name, in this case, .LOCAL.

The designers of mDNS made all attempts to ensure that it was not dependent on the Internet and the global DNS. Unlike the DNS and the other alternative naming systems given in this paper, mDNS is designed to be used only locally.

5.2 Tor

Tor is a collection of software that provides anonymity to its users by encrypting and routing their traffic through many different nodes over encrypted connections. Intermediate nodes in the Tor network do not know the origin or the destination of traffic in the Tor network. This is referred to as onion routing.

Tor is developed and maintained by the Tor Project.[24] It includes the Tor Browser, a modified version of Mozilla's Firefox browser that seamlessly allows users to browse both Tor sites and sites on the web.[25] Tor uses the top-level domain name .ONION for its naming and its own resolution protocol. .ONION is included in the IANA Special-Use Domain Names registry.[26]

---

[21] See Blockchain Naming Systems, https://www.blockchain-names.com/
[22] See RFC 6762: Multicast DNS
[23] See Bonjour, https://developer.apple.com/bonjour/
[24] See The Tor Project, https://www.torproject.org/
[25] See Tor Browser, https://www.torproject.org/download/
[26] See IANA Special-Use Domain Names, https://www.iana.org/assignments/special-use-domain-names/special-use-domain-names.xhtml

Tor provides increased privacy over the traditional TCP/IP protocol stack of the Internet. Partly because of this increased privacy it cannot use the traditional DNS and requires its own resolution protocol. Tor is an example of signaling to applications that a different resolution protocol should be used by using a distinguished TLD. .ONION names, referred to in the Tor protocol as addresses, are strings of either 16 or 5~~3~~6 base32 characters (e.g., 3g2upl4pq6kufc4m.onion) and are not human memorable.[27] So called "vanity" .ONION domains can be created by users by repeatedly generating names until finding one that is memorable. For example, Facebook previously used facebookcorewwwi.onion as their .ONION address. This lack of memorability can make Tor addresses difficult to use. A 2018 study found that Tor users have incomplete mental models of .ONION sites, as well as difficulty discovering, tracking, and authenticating them.[28]

The Tor Project has invested considerable energy in getting Tor to interoperate with other Internet services, including the DNS, the web PKI,[29] and other alternative naming systems. Some certification authorities already support providing X.509 web certificates to .ONION domains, and there is an ongoing effort to support .ONION with the ACME protocol.[3031,32] Web pages that both have a .ONION address and are available via the web can include the HTTP header Onion-Location to indicate the .ONION address that hosts their content.[33] There is even a project to get Namecoin names to work within Tor using .BIT.ONION names.[34] And the Tor Project is investigating ways to put addresses of Tor nodes in DNS resource records.[35]

Clients connect to a single node of the Tor network using the SOCKS protocol version 5.[36] Tor nodes connect to one another using the Transport Layer Security (TLS) protocol with self-signed certificates.[37] Both SOCKS version 5 and TLS use IP addresses as their underlying identifiers, but neither of them have a dependency on the DNS.

---

[27] See Tor Project Proposal 279, https://github.com/torproject/torspec/blob/main/proposals/279-naming-layer-api.txt
[28] See Winter, Philipp, Anne Edmundson, Laura M. Roberts, Agnieszka Dutkowska-Żuk, Marshini Chetty, and Nick Feamster. "How Do Tor Users Interact With Onion Services?," 411–28, 2018. https://www.usenix.org/conference/usenixsecurity18/presentation/winter.
[29] See Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates v2.0.0, https://cabforum.org/wp-content/uploads/CA-Browser-Forum-BR-v2.0.0.pdf
[30] See Internet Draft Automated Certificate Management Environment (ACME) Extensions for ".onion" Special-Use Domain Names, https://datatracker.ietf.org/doc/draft-ietf-acme-onion/
[31] See ACME for .onion domains, https://acmeforonions.org/
[32] See The 2023 Onion Plan: User Roadmap - Certificates, https://tpo.pages.torproject.net/onion-services/onionplan/roadmaps/certificates/
[33] See What ".onion available" means in my browser?, https://support.torproject.org/onionservices/onion-location/
[34] See Namecoin Resolution with Tor via ncprop279 and StemNS, https://www.namecoin.org/docs/tor-resolution/ncprop279/stemns/
[35] See The 2023 Onion Plan, Specs for DNS-based .onion records, https://tpo.pages.torproject.net/onion-services/onionplan/appendixes/dns/#implementation-considerations
[36] See RFC 1928: SOCKS Protocol Version 5
[37] See Tor Protocol Specification, https://github.com/torproject/torspec/blob/main/tor-spec.txt

5.3 Ethereum Name Service

The Ethereum Name Service (ENS) is a naming service based on Ethereum, a decentralized blockchain that allows the embedding of logic into its blocks to implement what are referred to as smart contracts.[38] One such smart contract in Ethereum implements ENS. It uses dot-separated labels like the DNS and is hierarchical, so a user controlling example.eth controls all domains beneath it as well. .ETH is a special TLD in ENS allowing any registrations, while other TLDs are restricted.[39] .ETH not registered in the IANA Special-Use Domain Names registry, nor is it in the DNS root zone.

As described in the Ethereum documentation:[40]

> The Ethereum Name Service (ENS) is a distributed, open, and extensible naming system based on the Ethereum blockchain.
>
> ENS's job is to map human-readable names like 'alice.eth' to machine-readable identifiers such as Ethereum addresses, other cryptocurrency addresses, content hashes, and metadata. ENS also supports 'reverse resolution', making it possible to associate metadata such as canonical names or interface descriptions with Ethereum addresses.

The primary use cases of ENS are to provide human memorable names for cryptocurrency wallet addresses and InterPlanetary File System (IPFS) identifiers, a distributed content distribution network where content is hosted by the users of the network.[41,42,43]

Once a user has reserved a domain in ENS, it can be used to reference a cryptocurrency wallet address. Cryptocurrency wallet addresses are used to store, send, and receive cryptocurrencies between users. ENS can be used to reference these addresses, thereby providing a human memorable identifier to remember and communicate them.

ENS can be used to reference content on IPFS. Content on IPFS is broken up into blocks of data that can be located at many different nodes in the network. This content is then addressable by a unique cryptographic hash, which is difficult to remember and communicate. ENS can be used to give a human memorable identifier to one of these hashes, so that people can remember and communicate IPFS hashes.

Because ENS is implemented in smart contracts that are embedded in a ledger, it should be impossible to remove names by anyone other than the private key holder. However, control of the smart contract that implements the ENS system and the .ETH namespace is handled by seven

[38] See ENS Documentation, https://docs.ens.domains/
[39] See ENS FAQ, https://docs.ens.domains/frequently-asked-questions
[40] See ENS Documentation, https://docs.ens.domains/
[41] See The InterPlanetary File System, https://ipfs.tech/
[42] See Trautwein, Dennis, Aravindh Raman, Gareth Tyson, Ignacio Castro, Will Scott, Moritz Schubotz, Bela Gipp, and Yiannis Psaras. "Design and Evaluation of IPFS: A Storage Layer for the Decentralized Web." In *Proceedings of the ACM SIGCOMM 2022 Conference*, 739–52. SIGCOMM '22. New York, NY, USA: Association for Computing Machinery, 2022. https://doi.org/10.1145/3544216.3544232.
[43] See "Why the Internet Needs the InterPlanetary File System - IEEE Spectrum.", https://spectrum.ieee.org/peer-to-peer-network.

individuals. Four of the seven must authorize any change to the smart contract through a form of multisignature authorization.[44]

The censorship resistance of ENS and most other blockchain based naming systems comes with an inherent downside for users. There is no entity an ENS user can contact if they lose the ability to make changes to their ENS names. There is nothing like a registrar, or other responsible entity to manage the relationship between a user and an ENS name. Every ENS user is individually responsible for managing their own cryptographic keys. If an ENS user loses their cryptographic keys the name can no longer be managed.

5.4 Unstoppable Domains

Unstoppable Domains is a for-profit company based in California, USA. They offer reservation of second-level domains in a select set of top-level domains. At the time of publication this list of top-level domains includes: .888, .BITCOIN, .BLOCKCHAIN, .COIN, .CRYPTO, .DAO, .NFT, .WALLET, .X, and .ZIL. None of these top-level domains are registered in the IANA Special-Use Domain Names registry, nor are they in the DNS root zone.

The Unstoppable Domains system is built on top of the Polygon blockchain platform, which is inspired by Ethereum and its computational model. Like ENS the primary use cases for Unstoppable Domains are to map names to cryptocurrency wallet addresses or InterPlanetary File System (IPFS) identifiers. Unlike ENS, Unstoppable Domains is a for-profit company requiring users to register names through their interface.

From blockchain-names.com:[45]

> Initial name registration is handled entirely through Unstoppable Domains and their web interface. The underlying contracts used for data storage [..] can only be initialized by Unstoppable Domains, forcing all users to register through Unstoppable Domains's web site and pay to register by directly paying Unstoppable Domains.

5.5 Gnu Name System

The GNU Name System (GNS) is part of the GNUnet system.[46] It promises full decentralization and name persistence. It uses the GNUnet Assigned Numbers Authority (GANA) registry hosted by the GNU project, and names are reserved on a first-come-first-served basis.[47] The GNS protocol is described in RFC 9498.

GNS has no concept of a root zone. Instead GNS uses the concept of a "start zone" that is configured locally and determines where to begin resolution. GNS users can instantiate names

[44] See Blockchain Naming Systems Ethereum Naming Service, https://www.blockchain-names.com/ens
[45] See https://www.blockchain-names.com/unstoppable
[46] See Wachs, Matthias, Martin Schanzenbach, and Christian Grothoff. "A Censorship-Resistant, Privacy-Enhancing and Fully Decentralized Name System." In Cryptology and Network Security, edited by Dimitris Gritzalis, Aggelos Kiayias, and Ioannis Askoxylakis, 127–42. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2014. https://doi.org/10.1007/978-3-319-12280-9_9.
[47] See The GNUnet Assigned Numbers Authority (GANA), https://docs.gnunet.org/gana/index.html

either under the .ALT TLD (See Section 9) or at the top level. In both instances, reserved names can be registered at GANA.

Since local users have complete control over their own start zone, every GNS user can potentially use a different namespace. Thus, there is no guarantee that names will be globally unique, or that a given name will resolve the same for different users. The only guarantee is that users with the same start zone will have the same view of the namespace. Essentially, every unique start zone defines its own namespace. This is similar in practice to DNS resolution using different root zones. The key innovation in GNS is to replace a search hierarchy with a distributed hash table that can include links to other hash tables.

GNS supports a special record type called GNS2DNS that facilitates integration between GNS and DNS.[48] GNS2DNS records contain a domain name and the name of a DNS resolver. When a client receives a GNS2DNS record in response to a GNS query, the client continues resolution of the name by sending a query to the DNS resolver included in the GNS2DNS record. In this manner the GNS2DNS record can delegate continued resolution of the name to the DNS.¶

6 The Importance of Referential Integrity to Internet Naming

For Internet technologies such as DNS or routing, there is top-down hierarchical delegation without a central authority.[49] What binds the system together is a set of common standards and assumptions that specifies the behaviors of and interactions among the components. One such assumption is referential integrity: regardless of who looks up a name, the response should consistently be what the domain administrator intended. If two people look up the same name they should both receive responses that are determined by the domain administrator (which may be different responses).

Or as the IAB put it in RFC 2826:[50]

> In the case of a public communications system [the] condition of a common symbol set with a common semantic interpretation must be further strengthened to that of a unique symbol set with a unique semantic interpretation. This condition of uniqueness allows any party to initiate a communication that can be received and understood by any other party. Such a condition rules out the ability to define a symbol within some bounded context. In such a case, once the communication moves out of the context of interpretation in which it was defined, the meaning of the symbol becomes lost.

Or as ICANN put it in ICP-3:[51]

---

[48] See RFC 94984: The GNU Name System, Section 5.2.2

[49] While the Internet Assigned Numbers Authority (IANA) is the global coordinator for the DNS root zone, there is no centralized authority for the global DNS.

[50] See RFC 2826: IAB Technical Comment on the Unique DNS Root

[51] See ICP-3: A Unique, Authoritative Root for the DNS, https://www.icann.org/resources/pages/unique-authoritative-root-2012-02-25-en

> The DNS is a globally distributed database of domain name (and other) information. One of its core design goals is that it reliably provides the same answers to the same queries from any source on the public Internet, thereby supporting predictable routing of Internet communications. Achievement of that design goal requires a globally unique public name space derived from a single, globally unique DNS root.

The global DNS started as an unambiguous referential system. The DNS, as originally conceptualized, was intended to create a reference from a single domain name to an IP address or other object. Until the administrator of the zone changed it, the reference would remain the same, and queries for the domain name would return the same set of IP addresses.

This basic behavior has been changing recently due to two trends in resolution. The first trend is the use of additional data associated with the query to determine the answer; instead of the queried name being the only thing that determines the answer, additional information associated with the query combined with the queried name determines the answer. EDNS client-subnet (ECS) is an obvious example of this trend.[52] With ECS, a recursive resolver includes the subnet of the querying client when it forwards the query to the appropriate authoritative server. By including this additional information the authoritative server is able to tailor its responses based on the location of the client.

ECS is by no means the only example of this trend. An authoritative server could give different answers when serving the same instantiation of a zone. For example, this technique is used to distribute traffic to different servers based on their current load, or perform steering of content requests within content distribution networks. Geolocation can also play a role here, and an authoritative server could tailor answers based on the conjectured geographic location of the querying recursive resolver. All of these examples modify the consistent relationship between a name and the set of IP addresses (or other data) it refers to. They modify the referential integrity present in the original conceptualization of the DNS.

Resolution based on associated data leads to ambiguity insofar as names are no longer sufficient to reproduce resolution outcomes. The same name can give different answers depending on when and where you query, what application you use, or what identity you authenticate as. As a result, an end user can no longer be sure of the context and outcome of the resolution of a name they see; however, delivered responses remain as intended by the domain administrator.

The second trend threatening referential integrity is the increasing use of domain names in naming systems other than the DNS. The global DNS is both a protocol and a syntax for domain names, and increasingly names using the ~~the~~ syntax of DNS names are being used in protocols other than the DNS protocol – alternative naming systems that make use of the domain namespace are increasing in use. Some of them use specific top-level domains (e.g., .ONION, .LOCAL) to signal that the resolution protocol should not be the DNS, and some of them do not. Sometimes the other resolution protocol can be determined based on the application they are

[52] See RFC 7871: Client Subnet in DNS Queries

used in, or from some other usage context. These names regularly leak into the DNS because applications attempt to resolve them via the DNS protocol, creating something of a mess.

However, as more alternative resolution systems are adopted it becomes increasingly difficult to determine which resolution system should be used to resolve a name and match the user's intent: the context needed for the appropriate resolution can get lost. As the resolution of a name is dependent on the knowledge of some further context, the systemic absence of such contextual information makes name resolution indeterminate. All names, including names in the DNS, are thus compromised in their utility.

¶

7 Perspectives on Ambiguous Internet Name Resolution

Everyone who uses the Internet depends on the domain namespace, but not in the same ways. This section discusses how two groups use and depend upon the domain namespace, and how its ambiguities contrast with the necessary clarities these groups require.

7.1 End Users

End users constitute the group of people who use Internet-connected computers. We are all end users in addition to any other roles we may have. Default settings in end user applications are important because most end users do not change default settings – many are not even aware that they can. The manner in which names are presented to end users is significant because this presentation forms the basis for how end users understand and interact with names in the applications they use.

After the invention of the world wide web and prior to the development of effective web search technology the DNS and domain names played an important role in how end users discovered Internet resources and navigated the web: a user looking for Starbucks would actually type "starbucks.com" and was directly aware that the domain name would lead the user to the desired content or service.

RFC 3986, written in 2005, had the following to say about the importance of URI transcription:

> The URI syntax has been designed with global transcription as one of its main considerations. [..] The goal of transcription can be described by a simple scenario. Imagine two colleagues, Sam and Kim, sitting in a pub at an international conference and exchanging research ideas. Sam asks Kim for a location to get more information, so Kim writes the URI for the research site on a napkin. Upon returning home, Sam takes out the napkin and types the URI into a computer, which then retrieves the information to which Kim referred. [..] A URI often has to be remembered by people, and it is easier for people to remember a URI when it consists of meaningful or familiar components.[53]

[53] See RFC 3986: Uniform Resource Identifier (URI): Generic Syntax

As search engine technology improved users relied less on domain names and URIs for ease of transcription and memory, and more on search engines for discovery. While the domain name is still used under the covers, a user looking for Starbucks now simply types “Starbucks” or even “latte” and a search engine leads the user to the desired content or service, with the user often unaware that a domain name is involved. As early as 2005 it was recognized that search engines were beginning to replace domain names as the primary means by which users accessed resources on the Web:

> As search engines have improved in user-perceived quality and ease of use, they have become a principal means of navigation to new destinations for many users. In place of guessing, intrepid Web travellers enter a descriptive word or phrase in the search engine and use the resultant list to direct their journeys. In June 2004, nearly 4 billion searches were conducted each month by almost 100 million people in the United States, an average of 33 searches per person per month. It appears that a consequence of the growing use of search engines for navigation across the Web may be a reduction, though by no means elimination, of the direct use of the DNS to support navigation by guessing domain names.[54]

This trend has continued, and end users browsing the web are increasingly unlikely to interact with domain names in the applications they use. Although still present in the background, a user may not know the website’s domain name, not remember it, not know how to spell it, or may prefer speaking instead of typing. Search engines solve these kinds of discovery problems much better than the alternative of requiring users to input a precisely written URI or domain name. Search engines are easier to use, more forgiving of end user mistakes, and adaptable to changing user input and circumstances.

Web browsers have adapted to be more user friendly in this regard by adding additional functionality to what used to just be an input field for URIs. Address bars have changed into general purpose input fields that can execute searches, interpret partial URIs, and execute user defined macros.[55] Instead of requiring a well-formed and user-unfriendly format for URIs, address bars now permit more free-form input and deploy heuristics to guess what the user wants. For example, a user could enter “coffee” into their web browser’s “omnibar” and be taken to the Starbucks website. This new kind of address bar that accepts more user-friendly input and commands is often referred to as an *omnibox*, or *omnibar*.[56] A 2021 study found that, “search engine functionality directly embedded in the browser has become a staple of user web

[54] See Council, National Research, Division on Engineering and Physical Sciences, Computer Science and Telecommunications Board, and Committee on Internet Navigation and the Domain Name System: Technical Alternatives and Policy Implications. Signposts in Cyberspace: The Domain Name System and Internet Navigation. Illustrated edition. Washington, D.C: National Academies Press, 2005, page 28.
[55] See Introducing Chrome Actions, a new way to navigate and take action right from Chrome's address bar, https://support.google.com/chrome/thread/83519363
[56] See Mozilla Developers Network omnibox, https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/API/omnibox

browsing", and that, "[..] the use of search terms in the address bar has eclipsed that of supplying URLs directly."[57]

Another reason why end users interact less often with domain names is the increasing use of mobile devices such as phones and tablets. These smaller devices have supplanted desktops and laptops as the primary means by which most end users interact with the Internet, and searching is even more important for discovery on small devices than on computers with large screens and full keyboards. Domain names are still in use, but end users seldom see them. It is still possible to enter a URI or domain name on a mobile device, but because it is cumbersome most end users do not.

In addition to the increasing dominance of search engines, other identifier technologies have emerged to compete for the end user's attention. In many geographies end users are more likely to encounter a quick-response (QR) code or a social media handle than they are to encounter a domain name. QR codes can be easily scanned using a camera on a mobile device, and then any interpretive context can be provided by the mobile device. The QR code may resolve to a URI, but this is different from using domain names for presentation that requires some interpretation by the user. The user sees a QR code, not the URI that the QR code resolves to, so while the global DNS is necessary to provide the Internet resource to the user it is, as is the case with using search engines, invoked without the user's knowledge.

For example, an English speaking user who encounters an IDN in Arabic may not have the knowledge necessary to understand or input that domain name. However, a QR code can be equally useful to both an Arabic speaker and an English speaker because their mobile device can intermediate on behalf of the user after the QR code has been scanned. Knowledge of a specific natural language or script is not required to use a QR code. All the user has to do is scan it and their mobile device does the rest. A downside of QR codes is that they are not memorable like many domain names. A mobile device with a camera is required to record a QR code for later use, whereas domain names can be more amenable to human memory.

Mobile devices also use dedicated apps to provide users with connections and services. Rather than visiting the Starbucks web site, many users will simply use the Starbucks app to find the nearest store, place an online order, and store their account balance for quick pickup and payment – again, hiding any use of domain names and any concept of name resolution.

End users are seeing fewer domain names in their physical environments than they were in the past. New technologies, especially search engines and mobile computing have changed the way end users discover resources on the Internet. There are also new competing identifiers to domain names like QR codes and dedicated applications. The global DNS provides a ubiquitous underlying default resolution context that QR codes and social media handles rely on to function. However, the end user does not need to understand how things work. From the perspective of end users, domain names are no longer the primary mechanism for finding resources on the Internet.

[57] See Crichton, Kyle, Nicolas Christin, and Lorrie Faith Cranor. "How Do Home Computer Users Browse the Web?" ACM Transactions on the Web 16, no. 1 (September 28, 2021): 3:1-3:27. https://doi.org/10.1145/3473343.

As part of ICANN Org's Affirmation of Commitments to the United States Department of Commerce, ICANN committed to organizing a review of new gTLDs shortly after they began being added to the root zone in 2013.[58] As part of this review, the ICANN Board directed ICANN Org to collect data as described in the Implementation Advisory Group for Competition, Consumer Trust and Consumer Choice (IAG-CCT) Final Report in February of 2015.[59,60] As a result, ICANN Org commissioned the Nielsen Company to conduct four global surveys. Four surveys were carried out in 2015 and 2016. They surveyed DNS registrants and consumer end-users. Registrants were defined as current domain name registrants at least 18 years old, and consumer end-users were at least 18 years old who spent a minimum of five hours a week on the Internet.[61] While focused on new gTLDs, the surveys also asked behavioral questions about QR codes, URL shorteners, and search engines.

All four surveys found that search engines were the predominant method for users accessing website content. This was consistent across both groups and both years. Registrants were slightly more likely to enter a domain name into an application than consumer end-users. URL shorteners did not see much significant use in either group and their use actually declined or stayed about the same in 2016. Only Africa reported significant use of URL shorteners with between 18-20% of all respondents saying they used them frequently. QR codes saw limited use, but their use increased among both groups in 2016. QR code usage was greatest in Asia, where in 2016 22% of registrants and 18% of consumer end-users reported using them frequently.[62,63,64,65]

There are numerous security-related effects stemming from the decrease in visibility of domain names, one of which is that it has made it easier for bad actors to defraud users. Deceptive domain names have long been around, and fooling users with names that look "similar enough" or are otherwise plausible – names such as chasebank.com (mimicking chase.com) or bankofthevvest.com and bankofarnerica.com (which can appear indistinguishable from bankofthewest.com and bankofamerica.com, respectively) – have long been common. When the user does not even see these domain names, the problem is worse, as a fraudster can register

[58] See Affirmation of Commitments by the United States Department of Commerce and the Internet Corporation of for Assigned Names and Numbers, https://www.icann.org/resources/pages/affirmation-of-commitments-2009-09-30-en

[59] See Implementation Advisory Group for Competition, Consumer Trust and Consumer Choice Final Report, https://community.icann.org/download/attachments/48349551/IAG-CCT%20Final%20report.docx?version=1&modificationDate=1418863127000&api=v2

[60] See Approved Board Resolutions | Regular Meeting of the ICANN Board | 12 February 2015https://www.icann.org/en/board-activities-and-meetings/materials/approved-resolutions-regular-meeting-of-the-icann-board-12-02-2015-en#1.e

[61] In Wave 2 the consumer end-user survey added teenagers (15-17).

[62] See Nielsen ICANN Global Registrant Survey, Wave 1, https://newgtlds.icann.org/sites/default/files/global-registrant-survey-25sep15-en.pdf

[63] See Nielsen ICANN Global Registrant Survey, Wave 2, https://newgtlds.icann.org/sites/default/files/global-registrant-survey-15sep16-en.pdf

[64] See Nielsen ICANN Global Consumer Research, Wave 1, https://newgtlds.icann.org/sites/default/files/global-consumer-survey-29may15-en.pdf

[65] See Nielsen ICANN Global Consumer Research, Wave 2, https://newgtlds.icann.org/sites/default/files/phase2-global-consumer-survey-23jun16-en.pdf

essentially any name with any registrar, fooling the user purely through the message or web site content, then abandoning those throw-away domain names as they get shut down. URL shorteners and other URL redirection services present additional privacy issues for users by introducing an intermediary that is sent data when a user clicks on a link.[66]

Very few users are aware that certain top-level domains can signal a different resolution context. For example, most Internet users are not aware that a domain name ending in .LOCAL or .ONION is intended to signal a switch to a different resolution protocol. ICANN OCTO maintains a listing of TLDs seen at the ICANN managed root server (IMRS) and their resulting magnitude.[67] It regularly features TLDs not delegated in the root zone in its top-twenty by magnitude list. APNIC maintains a list of top queries to Cloudflare's recursive DNS servers for names that do not exist in the DNS root zone.[68] For those TLDs used for alternative naming systems, appearing in these lists means that the domain name was not resolved via its "correct" resolution context. For example, a user may have input the domain name into an incorrect application.

7.2 Software Developers

Users expect that when they enter a name the "correct" thing will happen, but it is not always easy or possible for developers to determine what "correct" means. This sometimes requires developers of applications to guess what users want when they use a domain name or other identifier. The omnibox discussed in Section 7.1 is a good example of this. Users have the expectation that putting a malformed DNS name in an omnibox will take them to what they think is the correct website, or perform the correct action, for their individual sense of "correct". There may be no agreed-upon input syntax for omniboxes, but the web browser that is best able to guess what users expect will be best positioned to serve those users.

In most cases modern software developers interact with and use Internet identifiers through some kind of application programming interface (API), and it has long been the case with DNS that libraries have been available to resolve domain names without the programmer needing to care about the underlying protocol. These APIs abstract away, or hide, many of the details of the DNS or other identifier systems and allow the developer to focus instead on the unique problems their applications are intended to solve. Unless a developer is working on DNS software there is little reason for them to care how names in their applications get resolved. As the number and varieties of platforms, operating systems and programming languages have proliferated, so have the number and varieties of APIs.

[66] See Koop, Martin, Erik Tews, and Stefan Katzenbeisser. "In-Depth Evaluation of Redirect Tracking and Link Usage." Proceedings on Privacy Enhancing Technologies 2020, no. 4 (2020): 394–413. https://doi.org/10.2478/popets-2020-0079.
[67] See Welcome to the ICANN DNS Magnitude statistics page, https://magnitude.research.icann.org/
[68] See Delegated and NXDOMAIN requests through Cloudflare DNS, https://stats.labs.apnic.net/cfnxdata/

The default name resolution context for most Internet applications continues to be the DNS, descending from library calls released in 1983 on BSD UNIX,[69] but this will not always be the case. For example, JavaScript, the most popular programming language for the web, does not provide a function for resolving DNS names. JavaScript merely supports requests for URIs, and the web browser executing the JavaScript is responsible for resolving the domain name in the URI.[70] The JavaScript developer does not need to know anything about DNS resolution, even at the level of API calls: they only need to care about calling a function with a URI and handling the result. The browser may be performing DNS resolution itself, making a call to an API in the underlying operating system, or reading from its own internal DNS cache, and the application developer does not need to think about how that works. APIs also exist for alternative naming systems. Apple's API for their Bonjour protocol provides functionality for registering, browsing for, and resolving Bonjour services.[71,72]

When an application sees a name ending in .LOCAL it should use mDNS, not DNS,[73] to resolve that name, and similarly for Tor's use of names ending in .ONION.[74] It falls to the developer to be aware of these different resolution contexts and switch to the correct one when their software encounters a name that requires it. This context switching is not always easy, nor is it always clear at development time which resolution context a specific name requires or might require in the future. Even if the developer is aware of different resolution contexts, software is generally written in a way where every name encountered that looks like a domain name gets sent to the DNS for resolution. This then results in queries not intended for the DNS "leaking" into the DNS.

Sometimes it will be clear to a developer based on factors such as the use of a reserved TLD which naming system the developer should use to resolve a name. At other times it will not be clear which API to use for a name's resolution. Microsoft Active Directory uses both the DNS and what Microsoft calls Active Directory domain names. Depending on which API call is used to resolve a name it can be resolved using DNS resolution, or via Microsoft's Active Directory service.[75] Having the same object named differently in DNS and in Active Directory can result in an unpredictable name resolution process. This is why Microsoft generally recommends that

[69] The functions gethostbyname() and gethostbyaddr() first appeared in either 4.1cBSD or 4.2BSD UNIX. Both were released in 1983. See gethostbyname(3), https://www.unix.com/man-page/FreeBSD/3/gethostbyname/ and https://man.openbsd.org/gethostbyname

[70] See Mozilla Developer Network - XMLHttpRequest, https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest

[71] See Apple Developer - Bonjour, https://developer.apple.com/bonjour/

[72] See Introduction to DNS Service Discovery, https://developer.apple.com/library/archive/documentation/Networking/Conceptual/dns_discovery_api/Introduction.html

[73] See IANA Special-Use Domain Names, https://www.iana.org/assignments/special-use-domain-names/

[74] See The Tor Project, https://www.torproject.org/

[75] See Microsoft TCP/IP Host Name Resolution Order, https://support.microsoft.com/en-us/topic/microsoft-tcp-ip-host-name-resolution-order-dae00cc9-7e9c-c0cc-8360-477b99cb978a

names used in Active Directory and the DNS match (i.e., refer to the same computer or IP address).[76]¶

8 Implications of Ambiguous Internet Name Resolution

Internet names underwent a significant evolutionary change with the introduction of the DNS. Prior to the DNS, names were maintained locally (via /etc/hosts or similar) and they were essentially local aliases for the IP address of the service that was referenced by the name. This localized definition of names implied that addresses were the underpinning of integrity in the operation of the Internet at that time. If a packet was sent to the "correct" address then the response that this elicited from the remote service identified by that address and the specified port number was the “correct” one.

The DNS introduced a level of name coherence into the networked environment, such that any name-to-IP-address mapping was reasonably assumed to be consistent for all systems. If a transaction was directed to a given name, then the DNS would translate this to the associated IP address and any response was assumed to be authentic. This model formed the basis of the authenticity test in transport layer security (TLS).[77]

The address infrastructure of the Internet has been under considerable pressure, due in no small part to the exhaustion of available IPv4 addresses and lack of widespread adoption of IPv6. The consequent use of various address sharing models was intended to respond to this address scarcity, where the IP address was not consistently mapped to a single service.[78] The service name was cast into the role of service identifier, and the trust model in the authenticity of the service was built upon this framework.

Now, names, rather than addresses, are increasingly the foundation of trust for services on the Internet.

Ambiguous name resolution directly challenges this basis of trust. If the same name can resolve not only in different ways, but into different kinds of names (as with a non-DNS name) then this undermines our assumptions of trust in Internet naming more generally. The obvious response to this is to modify the trust credentials to include some form of context of resolution, so that the trust credentials are linked to the form of name resolution in order to resolve the potential ambiguity (See Section 4.1). However, such an approach has many pitfalls from a security perspective.

Namespace ambiguity can be used by bad actors as a mechanism to trick users into accessing resources that will harm them. The same name can resolve into different objects depending on the naming system used to resolve it. One of these could lead the user to their intended service,

[76] See Disjoint Namespaces, https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/plan/disjoint-namespace
[77] See What is SNI? How TLS server name indication works, https://www.cloudflare.com/learning/ssl/what-is-sni/
[78] See SAC079: SSAC Advisory on the Changing Nature of IPv4 Address Semantics

while introducing a man-in-the-middle, or redirect the user to a phishing site. Applications often attempt to resolve a name in the global DNS that does not exist in the global DNS, including for TLDs used in alternative naming systems, and these resolution failures are increasing.[79] This is a strong indication that sometimes applications resolve names using the incorrect naming system and get a response.

Recent research into blockchain naming systems used for malware command and control found that because it is currently more expensive for malware distributors to use current blockchain naming systems for command and control they continue to use the DNS.[80] Malware distributors are obligated to use centralized or semi-centralized infrastructure when connecting their malware to blockchain naming systems and defenders can block access to this infrastructure. For the time being, current general purpose blockchain-based naming systems are not attractive to malware authors for command and control because of their possible high cost, while naming-specific blockchains are easy for defenders to block because their centralized infrastructure is enumerable and relatively unchanging. This situation may change as naming-specific blockchains gain wider adoption, thereby making it more difficult for defenders to block access to them.

Namespace ambiguity can also lead to users encountering errors and getting frustrated at not being able to access the resources they expect. If a user encounters an error accessing Internet resources because their application used the wrong resolution context they will likely not even understand the problem or know who to contact. They may contact their Internet service provider or their IT administrator for help. However, even knowledgeable technical support staff may not be able to replicate the user's issue because they are unable to replicate the precise environment the user's application was using at the time of the error.[81]

As the SSAC discussed in SAC109, application developers sometimes choose to bundle DNS resolution with their application, thereby avoiding many of the issues inherent in namespace ambiguity.[82] By embedding the entire DNS resolution process in the application developers seek to avoid the problems inherent in deploying a single application into diverse resolution environments. DNS-over-HTTPS (DoH) has made this kind of bring your own infrastructure (BYOI) approach easier for application developers.

Names are increasingly becoming less conspicuous, less visible, and over time less a part of the user experience, making it increasingly difficult to signal anything to users with names. For example, a user scanning a QR code that contains a domain name will usually not see the domain

[79] See The Name Collision Analysis Project Study 2: Case Study of Collision Strings, https://www.icann.org/en/system/files/files/case-study-collision-strings-26jan22-en.pdf
[80] See Randall, Audrey, Wes Hardaker, Geoffrey M. Voelker, Stefan Savage, and Aaron Schulman. "The Challenges of Blockchain-Based Naming Systems for Malware Defenders." In 2022 APWG Symposium on Electronic Crime Research (ECrime), 1–14, 2022. https://doi.org/10.1109/eCrime57793.2022.10142131.
[81] See A possible example of a user experiencing a resolution error due to namespace ambiguity, https://social.technet.microsoft.com/Forums/en-US/06ae1597-6c28-4fd2-95ac-f52df6c72d34/microsoft-edge-doesn39t-open-pages-via-39microsoftedge39-protocol-and-netbios-name-as-a?forum=WinPreview2014General
[82] See SAC109: The Implications of DNS over HTTPS and DNS over TLS

name, so the user will be unable to determine that they have reached the "correct" intended service point.[83]

The same name can resolve in different ways (ambiguous name resolution), and names of service endpoints are less visible (names are less conspicuous to end users). It is the combination of these two issues that fundamentally threatens to undermine confidence in services on the Internet. ¶

9 Proposals to Facilitate Namespace Coordination

At the time of publication there are two proposals being considered by ICANN and the IETF to help facilitate domain namespace coordination.

The first proposal is SAC113: SSAC Advisory on Private-Use TLDs (See Section 1.1.3). The second proposal is RFC 9476: The .alt Special-Use Top-Level Domain.[84] Both of these propose setting aside portions of the namespace for specific purposes. SAC113 recommends a string be chosen that will never be delegated in the DNS root zone, the purpose of which is to facilitate private uses of the domain namespace. These private uses are intended to be for the DNS protocol. The concept underlying SAC113 is similar to that of RFC 1918 and RFC 4193 which set aside IP address spaces for private use.[85,86] At the time of publication ICANN is in the process of acting on SAC113's recommendation.

In contrast RFC 9476 proposes setting aside the top-level domain .ALT that is intended to be used for non-DNS protocols (i.e., Alternative Naming Systems). It adds .ALT to the IANA Special-Use Domain Names Registry.[87]

SAC113 and RFC 9476 are compatible with one another as their intended uses do not intersect. Both proposals reserve different names for different purposes and both are voluntary for operators to use. Neither proposes a registry to record uses or strings under their reserved TLD. If every designer of alternative naming systems followed the advice in these documents, many of the issues discussed in this publication would be mitigated. However, because there is no enforcement mechanism to ensure that they be followed, then there is no assurance that all namespace ambiguity will be eliminated. SAC113 and RFC 9476 are likely the best we can do to promote good behavior that minimizes downsides and risks.

In addition to the two proposals listed above, the ICANN organization has also been engaging in numerous activities that facilitate namespace coordination, including:

---

[83] See Major Energy Company Targeted in Large QR Code Campaign, https://cofense.com/blog/major-energy-company-targeted-in-large-qr-code-campaign/
[84] See RFC 9476: The .alt Special-Use Top-Level Domain
[85] See RFC 1918: Address Allocation for Private Internets
[86] See RFC 4193: Unique Local IPv6 Unicast Addresses
[87] See IANA Special-Use Domain Names Registry, https://www.iana.org/assignments/special-use-domain-names/special-use-domain-names.xhtml

- ICANN's strategic goals for 2021 - 2025 included an objective to, "Evolve the unique identifier systems in coordination and collaboration with relevant parties to continue to serve the needs of the global Internet user base."
- ICANN has hosted Emerging Technical Identifiers sessions at ICANN meetings.[88,89,90]
- The theme of the 2023 ICANN DNS Symposium was "The Integration of DNS with Emerging Technologies."[91]
- ICANN OCTO published OCTO-034: Challenges with Alternative Name Systems.[92]
- ICANN OCTO commissioned research into blockchain and other decentralized naming systems.[93]
- ICANN Launched the Special Interest Forums on Technology (SIFT).[94]
- ICANN facilitated the Name Collision Analysis Project (NCAP) in cooperation with the SSAC to investigate, and provide recommendations to help alleviate, name collisions.

The SSAC supports these and other activities of the ICANN Organization that facilitate namespace coordination.¶

10 Summary and Findings

This section contains a summary of the publication and includes the SSAC's Findings.

The default name resolution context for most Internet applications is the global DNS.[95] This mainstream use of DNS as the common resolution context helps ensure that domain names can be used as identifiers outside of computer interfaces. It also motivates designers of new naming systems to reuse the syntax of DNS names. This is discussed further in Section 3.

> **Finding 1:** Names that are syntactically equivalent to DNS names are being used in alternate protocols and different contexts. This is partly because applications written for DNS names easily support syntactically equivalent names, and also because users are already comfortable with this naming syntax.

¶

Whereas the original motivation for the design of the DNS namespace and name resolution protocol was to ensure that the DNS was viable for the technical capabilities of the time, we now

[88] See ICANN 58 Emerging Identifiers Technology Session, https://archive.icann.org/meetings/icann58/copenhagen/event/9nqD/emerging-identifiers-technology.html
[89] See ICANN 66 Emerging Identifiers Technology Session, https://archive.icann.org/meetings/icann66/meetings/NgGWYgp7jtE7Q2s8S.html
[90] See ICANN 78 Emerging Identifiers Technology Session, https://icann78.sched.com/event/1T4Id/emerging-identifier-technologies
[91] See 6th ICANN DNS Symposium, https://www.icann.org/ids
[92] See OCTO-034: Challenges with Alternative Name Systems, https://www.icann.org/en/system/files/files/octo-034-27apr22-en.pdf
[93] See Blockchain Naming Systems, https://www.blockchain-names.com
[94] See Charter for the Special Interest Forums on Technology (SIFT), https://community.icann.org/pages/viewpage.action?pageId=192220223
[95] See RFC 8499: DNS Terminology

see a number of new evolutionary pressures based on improved technical capability of the infrastructure: a desire to improve both speed and privacy of name resolution, and a desire to better verify the authenticity of DNS responses.

The DNS is the default naming resolution system for the Internet, and there is considerable operational infrastructure built around its administration and use. Thus, there are also pressures to resist change to the DNS and maintain the status quo in Internet naming. This is discussed further in Section 4.

> **Finding 2:** There are motivations to evolve Internet naming just as there are motivations to maintain the status quo. Wholesale replacement of the DNS as the default naming system for the Internet is very unlikely, therefore any successful alternative naming system must coexist with it for the foreseeable future.

¶

Everyone who uses the Internet depends on the domain namespace, but not in the same ways. A few users understand that certain top-level domains can signal a different resolution context, but most users are not aware of this. This is discussed further in Section 7.

> **Finding 3:** Relatively few users appreciate that a top-level domain can sometimes signal a change from the default naming system (i.e., global DNS). For example, many Tor users know that .ONION designates resolution via Tor, but very few users are aware of more than a couple of top-level domain names that signal a different resolution context should be used. It is likely that more will come over time.

¶

End users are increasingly unlikely to interact with domain names in the applications they use. Domain names and the DNS are still there, in the background, but most end users do not see or input domain names as they once did. A user may not know the website's domain name, not remember it, not know how to spell it, or may prefer speaking instead of typing. Search engines solve these kinds of discovery problems much better than the alternative of requiring users to input a precisely written URI or domain name. Search engines are easier to use, more forgiving of end user mistakes, and adaptable to changing user input and circumstances. Smartphone apps use domain names embedded within them, hiding them from the user. This is discussed further in Section 7.

> **Finding 4:** As domain names become less visible and less conspicuous, they become less a part of the user experience. Users still have a reliance on the underlying names to connect to expected services.¶

Names, rather than addresses, are increasingly the foundation of trust in the integrity of services on the Internet. Ambiguous name resolution directly challenges this basis of trust. If the same name can resolve not only in different ways, but into different kinds of names, then this

undermines our assumptions of trust in Internet naming more generally. This is discussed further in Section 8.

> **Finding 5:** Ambiguity in Internet name resolution can give unexpected results and therefore undermines trust in the integrity of services on the Internet.

¶

The Internet's permissionless innovation extends to developers of alternative naming systems. There is no requirement on designers of alternative naming systems to coordinate with other naming system designers. Therefore, name collisions between different naming systems are inevitable. This is discussed further in Section 8.

> **Finding 6:** Alternative protocols are increasingly using overlapping names. Therefore the same name will yield a different response depending on which resolution context is used. ¶

At the time of publication there are two proposals currently being considered by ICANN and the IETF. Both of these propose setting aside portions of the namespace for specific purposes. SAC113 recommends a string be chosen that will never be delegated in the DNS root zone, the purpose of which is to facilitate private DNS uses of the domain namespace. RFC 9476 reserves the top-level domain .ALT in the IANA Special-Use Domain Names Registry and is intended to be used for non-DNS protocols (i.e., Alternative Naming Systems). This is discussed further in Section 9.

> **Finding 7:** Mechanisms are being implemented at ICANN and the IETF to facilitate coordinated use of the domain namespace on a voluntary basis.

¶

11 Recommendations

Recommendation 1: The SSAC recommends that the ICANN organization continue to track and provide regular updates to the ICANN Board and community on both alternative protocols that make use of the domain namespace, and efforts to create mitigations and reduce risks inherent in the coexistence of multiple namespaces and protocols.

The SSAC recommends that the ICANN organization continue to keep the ICANN community abreast of new developments through such means as the Emerging Identifier Technologies panels that have been presented at a number of ICANN meetings.

~~**Recommendation 2:** The SSAC recommends that the ICANN organization continue to encourage collaboration and facilitate coordination among the various namespace communities.~~

12 Acknowledgments, Statements of Interest, and Withdrawals

In the interest of transparency, these sections provide the reader with information about aspects of the SSAC process. The Acknowledgments section lists the SSAC members, outside experts, and ICANN staff who co-authored or contributed directly to this particular document (Contributors) or who provided reviews (Reviewers). The Statements of Interest section points to the biographies of all SSAC members and invited guests, which disclose any interests that might represent a conflict—real, apparent, or potential—with a member's or invited guest's participation in the preparation of this Report. The Withdrawals section identifies individuals who have recused themselves from the discussion of the topic with which this Report is concerned. Except for members listed in the Withdrawals section, this document has the consensus approval of all of the members of SSAC.

12.1 Acknowledgements

The committee wishes to thank the following SSAC members and invited guests for their time, contributions, and review in producing this report.

SSAC Members
Jaap Akkerhuis
James Galvin
Russ Housley
Geoff Huston
Merike Kaeo
Warren Kumari
Barry Leiba
John Levine
Rod Rasmussen
Peter Thomassen
Tara Whalen

ICANN Staff
Andrew McConachie (editor)
Danielle Rutherford
Kathy Schnitt
Steve Sheng

12.2 Statements of Interest

SSAC member biographical information and Statements of Interest are available at: https://www.icann.org/en/ssac/members

12.3 Withdrawals

There were no withdrawals.