Report of the Technical Study Group: gTLD Integrations with Alternative Naming Systems

1. Background

The Domain Name System (DNS) (Mockapetris 1987a) (Mockapetris 1987b) is the global naming system for the Internet. The Internet Corporation for Assigned Names and Numbers (ICANN) Mission (ICANN, n.d.-a) invokes the need for uniform or coordinated resolution in the unique identifiers of the Internet. This is the reason for ICANN's role in the coordination of the root zone of the global DNS as well as policies that shape the overall operational environment of top-level domains (TLDs), especially those operated by contracted parties).

Since 2022, multiple generic TLD (gTLD) registry operators and potential registry operators have approached ICANN with requests for various kinds of integration of the global DNS with other naming systems. Now, ICANN is not responsible for every naming system connected to the Internet. For instance, consider login names to any website or social media system, or the local-part of an email address (the part that comes before the "@"). These operate, quite correctly, outside ICANN co-ordination. The distributed nature of the Internet prefers local operation by a network or service operator to global coordination, unless such global coordination is helpful for the operation of the network. This orientation towards local operation ("local" in all of the geographic, socio-political, and network-topological senses) is an important part of what allows the Internet to operate at global scale. That ICANN's role in global coordination is limited to cases where it is needed, not merely where it is possible, is reflected in ICANN's Bylaws, section 1.1. Yet, if the use of domain names that are in fact delegated in the global DNS begins, on a regular basis, to include functions that depend on naming systems that are not part of the global DNS, then the global coordination will itself be undermined to the extent that those alternative naming systems drift apart (e.g., in functionality, in responsible disclosure of functionality and interfaces, etc.) If there is to be a substantial integration of names from the global DNS into alternative naming systems, some coordination of how to perform that integration seems likely to be "reasonably necessary to facilitate the openness, interoperability, resilience, security and/or stability of the DNS." (ICANN, n.d.-a sec 1.1(a)(i))[1]

[1] It is worth noting in passing that, even if ICANN's responsibilities get extended to cover some naming systems that are alternatives to the DNS owing to integration of those naming systems with TLDs that are delegated in the global DNS, it does not automatically embroil ICANN in issues related to every

One mechanism that has been suggested by some registry operators relies on the domain name being “the same” in the global DNS and any other integrated alternative naming system. It is not hard to imagine other possible mechanisms for integration, but they are less clear and not widely discussed. The very fact that there are multiple requests along the “same name” line signifies that creating clear requirements for those designing new services using the same name would be helpful.

The term ”integration” for this purpose comes from an Internet Engineering Task Force Internet-Draft, “Integration of DNS Domain Names into Application Environments: Motivations and Considerations” (Sheth et al., n.d.). The draft discusses the use of domain names within applications, such as the way global DNS names were incorporated as the host-part of a URL in the early history of the web, or the way more recent social media protocols have integrated domain names within their systems for identifying their users. Within that context, there arises the matter of naming systems that are *not* directly using the global DNS, but that are nevertheless somehow linked to uses of domain names. In these cases, there may be reasons to create some kind of link from the global DNS into the alternative name system and its associated namespace. In keeping with the Internet-Draft, such a linkage may be called “integration of the global DNS with alternative naming systems.” Any case where a string exists as a label (i.e., of a TLD) in the root zone of the DNS, and where the operator of that TLD wishes to operate an alternative naming system using the same string, is in principle one of these cases.

Such integrations of global DNS gTLDs into alternative naming systems would constitute a registry service as described in the Registry Services Evaluation Policy (RSEP) (ICANN, n.d.-b), Section 1.1(c), because only the registry can ensure those operations happen within the global DNS.[2] In accordance with the RSEP, ICANN must evaluate such a proposed registry service for potential security, stability, or competition concerns. Accordingly, on 9 June 2026 ICANN announced that its CEO convened the Technical Study Group: gTLD Integrations with Alternative Naming Systems (often called “the TSG” in this report) to make that evaluation. Governing the scope of the effort was the group’s charter (https://www.icann.org/en/system/files/files/alt-naming-systems-tsg-charter-10aug26-en.pdf).

---

alternative naming system connected to the Internet. While it is almost inevitable that someone will call for ICANN involvement in every part of the Internet’s identifier infrastructure, ICANN’s remit need not expand beyond that “reasonably necessary” criterion of section 1.1(a)(i) of the bylaws. Just as ICANN need not have an opinion about the local-part of an email address even if email addresses use DNS names in the server-part, and just as ICANN need not express a view about whether an academic department’s machines should be named after moons of Jupiter or characters in *Les Aventures de Tintin* even when those machine names form the lowest-level label in a long domain name, ICANN need not be drawn into discussions of alternative naming systems except to the extent that they impinge on the global naming system.

[2] It is possible to operate alternative naming systems for individual names that appear in the global DNS without the involvement of the registry under which those individual names appear, or to operate an alternative naming system for a given TLD from the global DNS without the involvement of the registry of that TLD, or even to build an alternative root for the DNS without consulting ICANN. Such things exist on the Internet today, but do not work with the global reach and the scaling properties of the global DNS because they do not benefit from global coordination such as that organized through ICANN. In addition, the relationship to the RSEP noted here is part of the reason adding another naming system is best not done through Registry Voluntary Commitments. They are too close to the core of name resolution to be handled as voluntary commitments.

Over the past several years, and contemporary with the developments outlined above, research suggested that there are negative consequences for users arising from poor or abandoned integration of global DNS names with various applications on the Internet (Durand 2022) (Kaizer et al. 2024) (Kaizer et al. 2025) (Valapu et al. 2026). Yet, with pressures from blockchain-based systems (Hoffman 2024b) (Hoffman 2024a), there is little reason to suppose that all the naming systems on the Internet will converge towards the exclusive use of the global DNS. There is, therefore, a reason for registries, registrars, and ICANN to be concerned about how integration of the global DNS with alternative naming systems can work in a reliable, safe, and secure manner at a global scale.

1.1. The DNS Name Hierarchy, Alternative Naming Systems, and Human Factors

The global DNS is a hierarchical naming system, but in ways that might surprise someone not familiar with its operation. Any label boundary within a fully-qualified domain name might, but need not, represent a point of delegation within the global DNS. Moreover, and perhaps more surprising to someone not familiar with the system and its history, the hierarchy is only technical: in the name deep.hierarchy.example.net, there is no technical requirement that the policies at example.net reflect policies of net, nor that the policies at hierarchy.example.net reflect policies at example.net, and so on. This would appear to be at odds with the ordinary meaning of “hierarchy,” in which a corporate department’s policies (for instance) by and large include all the policies inherited from the division and corporation of which the department is a part. In any case, as of this writing there is no well-defined mechanism for retrieving (much less understanding) the prevailing policies within any given domain name. In practice, however, delegation-centric domains (like TLDs) usually enforce certain policies on the basis of what a subordinate domain is permitted to do while delegated, with the controlling sanction being that if the subordinate domain does not maintain the desired behavior, the delegation of the domain is removed. For instance, a domain name that does not maintain a sufficient number of nameservers enters a Hold state that pulls it from the global DNS. The addition of the required number of nameservers restores its delegation. That operating convention creates implications for the security properties of names.

In addition, it is not obvious that a hierarchical naming system is something natural or intuitive for most Internet users. Notionally, the scope of a name becomes more and more specific as one proceeds further from the root zone. In this way, “deep.hierarchy.example.com” is in principle somehow more specialized than “example.com”; and so it is reasonable to be more tolerant of experimentation at that deeper hierarchy. But instead of an awareness of hierarchy, over time many Internet users have come to understand a difference between “public suffixes” (Mozilla Foundation, n.d.) (names, like .com, in which other names may be registered for use) and everything else.

Because of the limited utility the hierarchical nature of the global DNS has demonstrated, some alternative naming systems may not be hierarchical. While for the purposes of this report’s analysis, the names may *appear* hierarchical like a global DNS name

(because this report deals with the same string, the names must contain what look like labels separated by dots), the hierarchy may not exist, may not function, or may be limited in a given alternative naming system.

It is already the case, then, that a number of relationships and distinctions within the naming system of the Internet (the DNS) may not be well-understood by or even known to Internet users, at least partly because the DNS does not make those relationships or distinctions technically accessible during resolution of names. And the global DNS has proven to be “good enough” for most purposes on the Internet despite being extremely old in Internet terms and despite not having features designed to facilitate upgrades. Nevertheless, the global DNS does not support all the things people might like to do with it, which is the source of pressure to integrate it with alternative naming systems. At the same time, if the various relationships and distinctions within the global DNS itself are not all apparent to Internet users, there is still less reason to suppose those relationships and distinctions will be apparent to users when more than one naming system is operating at the same time. This, ultimately, is the reason for caution when integrating the global DNS and other naming systems.

2. Problem Statement

This report was commissioned by the ICANN CEO in part because the RSEP requires preliminary review involving the ICANN organization (org) and possibly technical experts. It is therefore written primarily to provide advice to ICANN org in evaluating a proposed registry service when that service is presented for review.

This report addresses the integration of domain names with names in alternative naming systems. The domain names affected are expected to be principally top-level domains (TLDs, and see the next section for terminology definitions) in the global DNS, and second-level domains contained within the TLDs. It explores how to achieve such integration in a stable and secure way by using the same string(s) as the names in the global DNS, and depending on having the same party be in control of the global DNS name and any such integrated alternative-system name, with the registry functioning as the co-ordinating body to ensure the same party is in control of all the names. The report is written for the top-and-second-level case, but it would be trivially applied to other similar cases so long as the following conditions were met:

1. The (registry) operator of the service operates the parent level.
2. The operator of the service, and only that operator, operates the alternative naming system(s) (even if that operator then delegates some of its operational capabilities to others).
3. Domain names registered within the registry must integrate with the alternative naming system(s) (possibly by being excluded from the alternative systems).
4. The name in question is always the same name in every naming system involved.

It is important to emphasize that the mechanism explored by this report (with its corresponding limitations) may not be the only mechanism for integration from the global DNS into alternative naming systems. Nothing in this report should be construed as implying an endorsement of the particular mechanism of integration over others, a rejection of other

proposals for integration, or even an endorsement of the very idea of integration across naming systems. Through this report, the members of the TSG express their view that using the same name as controlled by the same party is unlikely to create significant security or stability issues within the meaning of the RSEP, assuming appropriate operational controls. The members do not assert that there is no risk, but believe it can be done safely and securely for the Internet with appropriate operational controls.

3. Terminology and Conventions in This Report

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 (Bradner 1997) and RFC 8174 (Leiba 2017) when, and only when, they appear in all capitals, as shown here.

Unfamiliar terms related to the global DNS in this report may be found in RFC 9499 (Hoffman and Fujiwara 2024) .

This report discusses the comparison of strings and names in the global DNS and strings and names in alternative naming systems. For the sake of brevity wherever clear, these will be referred to as **DNS-name**(s) and **alt-name**(s) respectively.

This report discusses the integration of two or more naming systems where the names in each system are the same string. See the discussion of The Same Name for what it means for **strings to be the same** as one another. In addition, this report discusses the case where these same strings in different systems are under the control of the same party. For brevity, integration on the basis of names both being the same string and being controlled by the same party is referred to as "**string+controller integration**." Within the report, whenever integration is discussed without additional context, it should be assumed that only string+controller integration is meant.

See The "Same Party" or Controller for discussion of what "**same party**" and **controller** means in the context of string+controller integration. Importantly for this report, for TLDs, the controller is usually called the registry operator (RO). For names registered within a TLDs, this controller is normally called the "registrant." Note that while some ROs use a Registry Service Provider (RSP) to perform technical operations on their behalf, the policy control of what happens within the registry is nevertheless the RO's responsibility and therefore this report treats the RO as the relevant party. When the report speaks of "the registry" or "the operator" as an actor, it is understood that the RO is responsible for the action being performed. Similarly, while registrants often or always proceed through a registrar, it is never the case that the registrar is the controller on behalf of the registrant.

This report is sometimes written as though there is only one alternative naming system in use along with the global DNS, but there is no logical requirement for that. Although there is only one global DNS, the same name might exist in 1..*n* alternative naming systems, and there is nothing in this report that precludes such an arrangement.

The term **"*n*LD"**, **where *n* is a number**, refers to the domain name of the *n*th level, where the TLD counts as "1" (and the root zone counts as 0). For instance, "example.com" is a

2LD below .com. (A second-level domain like this is sometimes seen abbreviated as “SLD” instead. Following other conventions, the document does not use 1LD or 0LD.) While the position of a label in a domain name is not technically significant, it often carries operational and policy implications that *are* technically significant.

The terms **security** and **stability** are used in this report according to the definition in the *Registry Services Evaluation Policy* (ICANN, n.d.-b).

3.1. Name States

The ICANN IDN Variant Issues Project, in its *Integrated Issues Report* (IIR) (ICANN IDN Variant Issues Project 2012), defined a number of states for domain names that are not directly useful for the purposes of this report, but that may be a model for state values for names under integration. In particular, the status terminology in the IIR is designed around individual labels, and always in respect of a given DNS zone. Because this report deals with different naming systems – some of which may not function one label at a time the way the global DNS works, and which are by definition not using DNS zones anyway – it is necessary to talk about whole names rather than only individual labels. In the IIR, the name statuses are related to a zone; in the present context the states are usually related to a naming system. Moreover, not all of the states in the IIR are relevant for the present discussion. To make clear that these values apply to whole names, each is prefixed with “name” in this document. Note that these are logical states, and may not be states that are reflected in any implemented system that is working to support the integration of multiple naming systems.

Because the terms “registration” and “registered” have technical, contractual, and legal meanings in the context of domain name registries, and because some of those meanings have financial implications, “registration” and cognates are only used in this report when referring to registration as it occurs in a pre-integration registry. That does not mean that integration cannot occur in such a registration system. It is instead that, whatever happens, the registration workflow occurs as it would in a pre-integration registry.

The terms that follow are in alphabetical order to aid later reference.

Name active (or activated): The name is functioning in some way within a naming system. A name for which answers are provided in every integrated naming system is **fully** activated and a name for which such answers are provided in only some naming systems is **partly** activated.

Name allocatable: The name is available to be allocated in the given naming system (see allocated, below). Note that a name that is available is allocatable in respect of every naming system, so a check that a name is allocatable does *not* show that the name has been allocated in any naming system.

Name allocated: The name is administratively associated with some entity in a given naming system for possible use within that naming system. This status does not imply that there are answers provided for the name in any naming system. Once a name is allocated in at least one system, it needs to be allocated or withheld in all of them (but that action might not be completely automatic).

Name available: This state does not have a corresponding entry in the IIR. This state is for a name that could be enrolled but that is not actually enrolled. If it were enrolled, it could then be allocated in at least one naming system.

Name blocked: This state is for a name that is not permitted to be enrolled and therefore not permitted to be allocated within *any* given naming system.

Name deactivated: The opposite of activated. Normally, this term is used with respect to any one naming system, and so the name is deactivated with respect to that system, and so is really **partly** deactivated. There is a separate term, disabled, for a fully deactivated name.

Name disabled: This state does not have a corresponding entry in the IIR. This state describes a name that is enrolled and allocated in at least one naming system, but that has been removed from service for some reason.This might include administrative removal under the order of a court, for instance. A disabled name is, in effect, one that is deactivated in all the integrated naming systems.

Name enrolled: This state does not have a corresponding entry in the IIR. This state is added here as a distinction in order to emphasize a basic logical requirement for this kind of integration, and enrollment often happens implicitly (as when, for instance, a name is registered in a registry for the global DNS). This state describes a name that is known in some way to the collection of integrated naming systems and that is associated with a controller. This enrollment state is, logically speaking, a necessary condition for allocation within any naming system, but might happen as the *result* of such allocation within one naming system. It is *not*, however, relative to any one of the naming systems under management, but rather applies to the entire collection of integrated naming systems.

Name suspended: This state does not have a corresponding entry in the IIR. This state describes a name that is enrolled and that is activated in at least one naming system, but for which various administrative updates are prohibited pending the resolution of some issue. In existing ICANN practices, Uniform Rapid Suspension provides an example of this state. Unlike a name that is disabled, the name might continue to function within the naming system in question. This term is defined such that it is logically possible for a name to be suspended in a given naming system, but it appears more likely that a name would be suspended in all naming systems at once (and so a discussion of a name suspension without noting a naming system can be presumed to mean all naming systems at once).

Name withheld: The name in a given naming system is set aside for possible allocation to some entity. In this strict sense, a withheld name is not actually allocated to anyone.

4. Basic Elements of String+Controller Integration

4.1. General

As a general matter, for string+controller integration, naming systems are integrated if and only if a name that is made up of the same string in each such system is always either controlled by the same controller or else withheld for the exclusive possible control by that same party. These two criteria MUST always be reliably satisfied for the naming systems to be integrated. If the criteria *are not* reliably satisfied, the naming systems are not integrated; and if the criteria *cannot* be reliably satisfied, then the naming systems and their associated namespaces SHALL NOT be treated as candidates for integration.

Because this string+controller integration is an integration of the global DNS into the alternative naming system, it requires that the names in question must meet minimum requirements for conformance at all times. This section discusses the requirements themselves. The following two sections (Enrollment Via Registration in Shared Registration System and String+controller Integration in a Distributed Mode) outline two different models for how the requirements might be met.

4.2. The Integrating TLD

In the case of the TLD itself, the string MUST be the same and the controlling entity MUST be the registry operator of the TLD. If the registry operator does not control the use of the name in the alternative naming system, then the name is not a candidate for integration under the string+controller model. Otherwise, this requirement is met because if the registry is in fact in control of all the name systems answering for that same string, the necessary conditions for integration under this definition are provided.

4.3. Integration of Subordinate Names

An integrated domain might adopt one of three rules for names registered within it: must-integrate, may-integrate, or integration-prohibited. It is difficult to imagine a delegation-centric zone (like a TLD) that would want to integrate the top-level name itself, and yet would not require or at least permit integration at delegations underneath itself. Assuming, then, that an operator does not prohibit integration of subordinate names, then the allocation of a name in any of the naming systems MUST either withhold or allocate that same name in (all) the other system(s) as well, and only to the same controller. We may call this function of establishing the possible allocation of a name in at least one of the integrated naming systems “enrolling the name.” For string+controller integration, an enrolled name MUST be the same string in each relevant naming system, and it must be controlled by the same entity in each such system. There are multiple possible ways to achieve this result. For the sake of brevity and comprehensibility, this report boils them down to two cases: using a single Shared

Registration System (SRS) registration as an enrollment controller or using data distributed across several authorities to control enrollment.

4.4. The “Same Party” or Controller

Regardless of how alike or different the naming systems involved are, or how names may be allocated to controllers within those systems, the string+controller model requires the association of the name with a controller, and that controller of the name MUST be the same party[3] across all naming systems.

These issues are solved in different ways in the two following models, but in either case, the requirement for an identical controller of the identical string MUST be met. If it cannot be, the domain is not a candidate for string+controller integration.

4.5. The Same Name

There are three ways to determine whether a DNS-name and an alt-name are to be regarded as the same string. For the DNS-name, this procedure MUST be conducted one label at a time, which effectively means label-at-a-time treatment for all other integrated systems as well, even when those other naming systems do not actually work label by label. In other words, if the other naming system does not operate on labels, the name in the alternative naming system will be processed as substrings of the whole name.

The first way is when the label is neither a U-label nor an A-label (Klensin 2010). In that case, the string is the same if it would match under the matching rules of STD 13 (Mockapetris 1987a sec 4.3.2, 1987b sesction 2.3.3). Informally, labels match if the DNS-name label and the alt-name label (or substring) match using a case-insensitive ASCII comparison.

The second way is when the label as encountered[4] is an A-label. In this way of matching, the string is the same when, in the alternative naming system, it is the same string under the matching rules of STD 13, as above; or in case the string, wrapped to lowercase and NFC-normalized (Unicode Consortium 2025), and then converted to an A-label, it is the bitwise-equivalent string of the A-label. Informally, this might be restated, “If the DNS label is an A-label, then the corresponding part of the alt-name must be either the matching A-label or convertible to the corresponding matching U-label; otherwise, it is not the same string.”

The third case is where the label as encountered is a U-label. In that case, the string is the same when, wrapped to lowercase and NFC-normalized, it is bitwise-equivalent to the U-label; or, if converted to U-label form, it is the bitwise equivalent of the U-label. Informally, this can be restated, “If the DNS label is a U-label, then the corresponding part of the alt-name must either be the matching A-label, or else convertible to the same U-label.”

Finally, the name is the same if and only if, after whichever of these procedures is used for each label, every such label in the DNS-name matches the corresponding substring of the name in the alt-name, and the alt-name contains a dot (“.”, U+002E FULL STOP (The Unicode

[3] The party is either a legal or natural person.

[4] “As encountered” because, formally, every A-label is exactly one U-label, and conversely. So, from a formal point of view, cases two and three are the same; but are covered here separately in order to ensure there is no dispute about the form of the name.

Consortium 2025)) at the boundary of each label of the DNS-name in presentation or common display format (depending on which is in use for the DNS registry).

4.6. Establishing Registry Control of Alternative Systems

As a general matter, the question of how to establish the global DNS registry operator's control of an alternative naming system is at least as much a contractual as a technical one. For the purposes of evaluating whether a given global DNS registry might be integrated with some alternative naming system, it is at the very minimum required to show that the entity controlling the global DNS registry also has effective control over the same name in the target alternative naming system. It MUST be able to provide all relevant control guarantees over any alt-name corresponding to an integrated DNS-name such that the alt-name cannot drift from the DNS-name in terms of string or controller. Otherwise, the "same entity in control" criterion of string+controller integration fails, and the name is therefore not a candidate for integration via this method. The most obvious way of demonstrating the relevant effective control over the alternative naming system is to prove exclusive control over that naming system.

To establish the control by the operator over the alternative naming system, the operator might provide the details of every interface for introducing changes into the alternative naming system, and then to demonstrate that the operator is the sole entity able (either of necessity or due to permissions and access controls) either to use all such interfaces or to authorize the events those interfaces process. This kind of demonstration proves the registry operator is in control through demonstrations of access controls.

Another means to establish such control would be to provide proof that the system can only process changes in the presence of change authorizations that can be issued only by the registry operator (such as a cryptographic token that authorizes the change, with demonstrations of appropriate validations). This would be a demonstration of the system's responsiveness to authorizations by the registry operator and resistance to authorizations issued by other parties. In the case that the operator might not have exclusive control over the entire alternative naming system, an operator would at least need to demonstrate the ability to deny all other possible controls over any alt-name corresponding to an allocated dns-name whenever that dns-name allocation was not also going to change in a corresponding way.

It is not possible, in general, to specify how to undertake an evaluation of possible controlling entities without details of the target naming system because there are too many possible ways for a naming system to be updated, given the range of things such systems may be trying to accomplish. At the same time, the overarching principle must be that the controls need to be sufficient to guarantee that an integrated name cannot drift in respect of the string or controller guarantees across any of the integrated naming systems. The stronger that guarantee is (and the easier it is to understand the ways in which the guarantee is true), the better the case that integration of the naming systems is safe and secure. This case ought to be convincing not only to an expert evaluator, but needs to be made comprehensible more generally so as not to undermine confidence in the security and stability of the Internet's identifier systems on the part of the Internet's users.

5. Enrollment Via Registration in Shared Registration System

In contemporary global DNS operations, names within a TLD are registered and maintained through the shared registration system (SRS). The SRS enforces the technical and policy conformance of names registered through it. The controlling entity of a domain name registration is the “registrant.”

That model of registration might be extended to other naming systems by, in effect, adding alternative naming systems as “plugins,” allowing the SRS to populate not only the global DNS naming system, but any other naming system for which the SRS has the necessary data. This is likely the easiest and most straightforward mode of integration of the global DNS with an alternative naming system. This approach to integration effectively brings the integrated alternative naming system(s) under the umbrella of any policy that would otherwise apply to global DNS names under prevailing ICANN policies.

5.1. Policy and Operating Convention Coherence

Because this approach to integration simply adds a new naming system on to the back end of the existing operating registry, any registry that conforms to existing ICANN policies and the prevailing operating conventions of the registry market automatically can provide the same functionality within the additional naming system(s), *provided that* the new naming systems do not themselves introduce mechanisms by which those policies or operating conventions are undermined. This caveat is significant. Under the Registry Services Evaluation Policy (ICANN, n.d.-b), it is still necessary to evaluate a given naming system that will be added to ensure that it does not itself introduce security or stability issues. The exercise of demonstrating that the resulting system does not harm the Internet is ICANN’s overarching duty in evaluating the registry service. The registry applying to offer such a service MUST demonstrate that the alternative naming system does not itself represent a risk to the security or stability of the Internet.

Assuming no such harm, then it follows that any operation on a name in the registry might be extended to the alternative naming systems. This requires some means to manipulate the registry along those extended lines. Extension(s) to the SRS commands for manipulating registrations (for contracted parties, this is the Extensible Provisioning Protocol, EPP (Hollenbeck 2009c)) are likely to be needed to permit operations on the alternative naming system.[5] Depending on the technology in question, it might be possible to create an extension either to the Domain Name Mapping (Hollenbeck 2009a) or else to create a completely different object mapping if that is called for. In the second case, the mapping MUST provide

[5] It is logically possible that such extensions might not be needed in case there are no optional components in use in the alternative naming system and every operation on a DNS-name entails exactly one corresponding identical operation on an alt-name, and there are no other operations to be performed on such an alt-name. It is hard to imagine what work the alternative naming system would be doing in such a system, but the possibility is noted for completeness.

means to ensure the object is related to a domain object in the EPP repository. It is hard to evaluate whether a different object mapping would meet the string+controller integration requirements, and it seems very much preferable to cover this need through extensions of the Domain Name Mapping, but the relationship between domain and host objects (Hollenbeck 2009b) under EPP provides a model for how alternative objects might relate to domain objects. In addition, a work in progress (Galvin and Bauland, n.d.) provides a promising mechanism to ensure two registry objects are always treated as one.

Registries currently provide information about data in the registry through Registration Data Directory Services (RDDS), normally provided via the Registration Data Access Protocol (RDAP, (Newton et al. 2015a, 2015b; Hollenbeck and Kong 2015; Hollenbeck and Newton 2021a, 2021b; Blanchet 2022)). Extensions to RDAP necessary to access the new data in the registry MUST be specified. Even in the logically-possible case that no EPP extensions were needed, it seems impossible that no RDAP extension would be needed since the alternative naming system will by definition contain distinguishable data about which there may be queries that need answering. RDAP extensions for use with alternative naming systems MUST be published and MUST be registered in the Internet Assigned Numbers Authority (IANA) RDAP Extensions registry as defined in RFC 7480 (Newton et al. 2015a). Re-use of extensions, including extensions not produced by the registry service applicant, is RECOMMENDED.

Some likely names for integration appear to be names that are already in operation in alternative naming systems *but not* in the global DNS. Such naming systems may have varying degrees of information about the controller of the names. Because this model reduces enrollment to registration in an SRS according to the current prevailing model for global DNS registration, it requires registration in the SRS, including information about the controller (the registrant, for SRS purposes) that may not be currently available. To undertake such an integration would require that missing data be provided. The registration of names in the SRS, some of which might not be intended ever to be candidates for entry into any global DNS zone, may have cost implications for the prospective registry operator that are beyond the scope of this report. In addition, owing to usual restrictions to do with the SRS for domain names, it may not be possible to register names that do not conform to the requirements or conventions for global DNS operation. For instance, this model cannot accommodate the provisioning of non-global DNS names that contain more than 255 octets (due to limitations of the DNS protocol) or names that contain emoji characters (due to the absence of normalization forms for emoji characters in Unicode).

It is worth emphasizing that this model of integration supports not only the basic registration model, but also implies the registry and all the potential registrants and users of registrations under the registry have the technical and operational capability to support all existing ICANN consensus policies throughout the integrated naming systems, even for names that are not activated in DNS. This approach to integration effectively brings the alternative naming system(s) so integrated under the umbrella of any policy that would otherwise apply to DNS names under prevailing ICANN policies.

This model for meeting the requirements for string+controller integration offers a simple, functional integration from the global DNS into another naming system, with the advantage that policy and protocol constraints already provided by an existing SRS automatically extend to the

new naming system. This comes at the cost that every name in the alternative naming system must *also* be at least a candidate name for the global DNS registry.

6. String+controller Integration in a Distributed Mode

Enrollment of a name into an integration of the global DNS and other naming systems does not have to depend on a single SRS. Instead, enrollment might happen in a distributed system that provides the necessary proof that the string is the same in all the relevant naming systems, and that the same string is always controlled by the same entity. This mode of operation might seem more natural in particular for naming systems related to decentralized and distributed-operation systems such as distributed ledgers or blockchains. For some such systems, settlement and closure can sometimes be uncertain. By contrast, because of the ever-advancing zone serial number (in the DNS SOA resource record required at the apex of every zone, and because this is ultimately about integrating the global DNS and other naming systems), it is necessary to be able to state with certainty that the string+controller at a given point in time indeed matches across the naming systems, which shows that the integration is working. The proof that the criteria for string+controller are reliably satisfied is a logical construct usefully called the Unified Source of Truth (or USoT).

It should be noted before proceeding that the discussion below considers matters generally from a technical point of view, in keeping with the scope of the TSG's charge and the expertise of the members (which leans more heavily towards the technical, and without deep expertise in contract law). But it must be recognized that technical means are not the only ways to achieve some of the assurances that are needed for string+controller integration. Many assurances on which registration markets depend today are achieved through legal means rather than through technical facilities that are built to provide the necessary evidence that a given condition prevails. This report largely avoids talking about contractual ways to achieve the stated goals because the group that produced this report is a *Technical* Study Group and not a Legal Study Group. Nothing in the remainder of the report should be taken to suggest that TSG members do not recognize the way law and contracts can be used (and are in fact used today) to provide certain necessary assurances.

6.1. The Unified Source of Truth (and Consequences)

Given the requirement that the different naming systems match on multiple criteria, the registry wishing to provide integration of different naming systems MUST ensure that there is one, eventually-consistent source of truth. In a traditional system design, this single source of truth would be a backend database of some description.

But in a system of different but integrated naming systems, it may not be practical to use a single back end system to store all the data. In that sense, rather like the global DNS itself, the "database" is found in parts across several systems. The requirement is only that

modifications of that single source of truth are executed in a way that ensures eventual consistency for each change, such that the fundamental guarantees required for string+controller integration are always provided for in all the resulting systems. The key difference is between a traditional back-end data store and a single, eventually consistent, USoT that can be distributed in its operation, in the instantiation of the nodes, and even in the authority over various components that make up the data store. The architectural implications of this difference can be found following in Understanding the USoT by Example.

Recognizing the overall system as providing a single, eventually-consistent USoT makes the claim of having truly separate naming systems a little bit softer. None of the naming systems can be truly fully independent of one another because they all rely on the USoT for their data. Similarly, there is no logical possibility that, if the USoT in fact undergirds all the name registrations (and it is in fact a unified source), matching strings could end up allocated or reserved to different entities in different naming systems. Every name in any of the underlying naming systems MUST be under common control (even if that control is distributed), which effectively means that all the naming systems are under a kind of coordination even as they appear to operate in quite different and possibly only loosely-related ways. The name spaces managed in these different naming systems form part of a single space that is fully managed.[6]

Proving the USoT works, is in fact unified, and is in fact a source of truth, is dependent on the underlying naming systems' properties and the ways they may be audited. The complexity of such proofs rises as the number of naming systems rises, especially if each such naming system works in a different way. In other words, operating many disparate naming systems appears to increase the risk to the overall integration due to complexity.

6.2. Same Controller

One might object that distributed operation means a registry operator does not have information about the registrant of a name, or that the normal operation of a given naming system does not normally associate registrant information with the name in question. But rather than being an objection, this illustrates how a distributed system providing the source of truth works. Most TLDs operate with registrars working as intermediaries between the registry and registrant. The registrar must, of necessity, know the identity of the entity with whom they are transacting business. The registrar might be treated as the authority on the matter and be held to various contractual terms that ensure the registrant is the same. Or else, the registrar would be in a position to provide an opaque but durable identifier of the registrant for the name to the registry or anyone else who wants it. This might even be a cryptographic identifier allowing the registry or someone else to validate the identification of the registrant without revealing the actual identity of the registrant. Of course, such an identifier need not be generated by a registrar – it could be a wallet, for instance. The fundamental requirement is only that any registration event of any DNS-name or alt-name of the same string can reliably check that string is being registered by the same entity. That proves the string+controller integration is in place.

[6] The skeptic should remember that many people have gone out of business waiting for the Internet to break down at last, simply because there is no party in central control of it.

The ability to check such conformance does not entail that it need be a real-time operation. For instance, a large blockchain network with a long settlement period could have contending transactions that made it into the blockchain and that are not yet settled when an EPP command is received at the global DNS registry component of the integrated systems. Depending on the kinds of transactions that might be contending, it is possible that the DNS registry command processing must be held in a pending state until other elements of the naming system have settled. In this case, the DNS registry command could finish processing, with the underlying object held in a pending state until the various other conditions are met, at which point processing may complete for the global DNS as well. As a result of this possibility, there are likely some conditions into which the DNS registry can enter when waiting on another naming system that would exceed the usual processing experience for a DNS-only registry. Such variance seems likely to happen even with synchronous operation when multiple naming systems are involved. Existing committed service levels for existing registries operating with fully centralized environments are quite generous and easily met, so one might expect a fully distributed operation, such as the kind being discussed here, to be able to meet similar commitments.

Given the emphasis on distributed operation above, one might argue that this account of “same party” need not extend to the point of *natural or legal entity*. Control could be functional control by multiple legal entities under some kind of agreement among them. But in this case, an analogy with registry operators more generally is worth considering. A registry operator need not perform all of the functions to which it is contractually obligated itself. It was never expected that such operators would design their own hardware for registry operations, or hand-wrap their own core memory modules. In the contemporary network, it is common for network service operators to put those services in others’ premises (whether that be leasing space for a set of owned racks within a data center, or running services in a provider’s cloud). And of course, some registry operators do not perform registry technical services at all, contracting those to a registry service provider. A registry operator is nevertheless responsible for the performance of its registry, and neither registrars nor ICANN need to have separate contractual arrangements with all operators of individual systems on which the registry depends. Contrast this need for the responsibility to be held by a single legal entity with the way the USoT need not be instantiated in any single database or even under any single system operator’s control. The importance of the single legal entity then becomes plain: without the requirement, an entity might attempt to shed responsibilities it has onto the operator of another service. With the requirement, the controller can expect technical mechanisms to prove the controller satisfied its obligations even within a system of distributed operation and data control.

6.3. Proving Control

At the same time, the conceptual notion of the USoT means that names in the manifold naming systems cannot (as a logical matter) fall out of sync with one another without violating the principle that they flow from a unified source of truth. This guarantee can be used to avoid the requirement for a large quantity of contractual provisions intended to be monitored for compliance because they all boil down to the constraint that changes cannot proceed through to any naming system without either being available in the other active naming system(s), or

else being reserved from activation. A prospective registry service operator applying to operate such a registry service and prove its claims via technical tests must demonstrate, above all, that such controls are in place and are effective not so that all the participating parties *can* produce string+controller integration if they wish to, but rather, that the participating parties *cannot break* the string+controller integration without the instruction or approval of the controller. Precisely how to demonstrate this property is not something that is possible for the present report to state because what is being described is a genus of operational designs and not a specific design. It is difficult even to start listing classes of examples because the combinatorial effects of combining different operating models means that, even if the number of options for operational choice were fairly small, the total number of possibilities that would need to be considered could grow large. Prospective operators of such a registry service, however, would need to demonstrate convincingly why and how that property could not be violated under deployment conditions. Otherwise, the service would not be a service to integrate operation of the global DNS and one or more alternative naming systems, but would instead be a service that could lead to confusion, violated expectations, and fraud.

6.4. Blockchain, Yes, But Thin Registries Too!

The discussion above may make the project of distributed operation seem huge and much too complicated to solve in support of the prosaic problem of registry management. Yet there is an existing, deployed technology that operates in this distributed way and that, if considered purely from the technical perspective, shows the USoT in action. It may also provide evidence of the satisfactory control that contracts provide.

Some domain name registries, already today, are referred to as “thin,” meaning they have no information about the parties responsible for a domain name. Instead, only the registrar has that information. This means that while the registry is in a position to assert that a given string is used as the same name in two naming systems, the registry is not itself in a position to assert they have the same controller. This is because technically the controller is known only to the registrar, and the only standard query mechanism to retrieve such information from a registrar is RDAP, which cannot share a transaction scope with the registry database. (While the registry and registrars are also in contact via EPP, the transaction framework of EPP is generally client-server, with the registry being the repository or server.) Worse, if the registry has implemented the DNS-name and alt-name integration using separate object mappings, the identity of the controller for the name needs to be associated with two database objects, which have to succeed or fail at once. They also have to refer to a common object but that object is not in the local database, so presumably, cannot be referred to within a single transaction. This is an example of a case where the mere introduction of global DNS integration adds a wrinkle to move a normal fact of global DNS registry operation towards distributed operation. One way to make sense of the registry’s state is to use the idea of the USoT.[7] (Of course, as noted previously, another answer is to depend on contracts.)

[7] The introduction of the “Same Entity Principle” from a work in progress (Galvin and Bauland, n.d.) might appear to obviate the USoT. In fact, it is an example of it in action, where the introduction of a means to

6.5. Understanding the USoT by Example

This section is meant to give those unfamiliar with distributed systems operations a brief and simplistic idea of how the USoT system might be expected to work while still providing certain guarantees of data integrity and consistency and so forth. Because it is intentionally simplistic, those experienced with distributed systems may not find it too informative.

A Relational Data Base Management System (RDBMS) normally offers the so-called "ACID" guarantees:

- **Atomicity** is the property in which each transaction either succeeds completely, or fails.
- **Consistency** is the property in which any transaction always maintains a valid state. As with atomicity, if a transaction produces an inconsistent state, the transaction must be rolled back.
- **Isolation** is the property in which two transactions do not affect one another. Isolation can be exceptionally important under high contention loads because it can prevent a transaction from completing if a different transaction has changed the conditions the current transaction was depending on.
- **Durability** is the property in which, once a transaction has completed, any changes that it made will persist even if the system crashes immediately at the moment of the commit.

Some practical results of a traditional representative registry architecture can be seen in Figure 1. The diagram illustrates how there are clear boundaries of what is inside the registry's scope and outside it, and also how the registry naturally divides into the data plane that accepts writes and the plane that does not. The distributed nature of the RDDS, however, is clear here: part of that data can be retrieved from the registry, but other data needs to be retrieved from the registrar. Finally, we can see that registries sometimes use external parties to deliver authoritative global DNS data, so those servers are shown as partly hanging outside the registry's operational control.

treat multiple objects as having an important relationship to one another makes the distributed system easier to use.

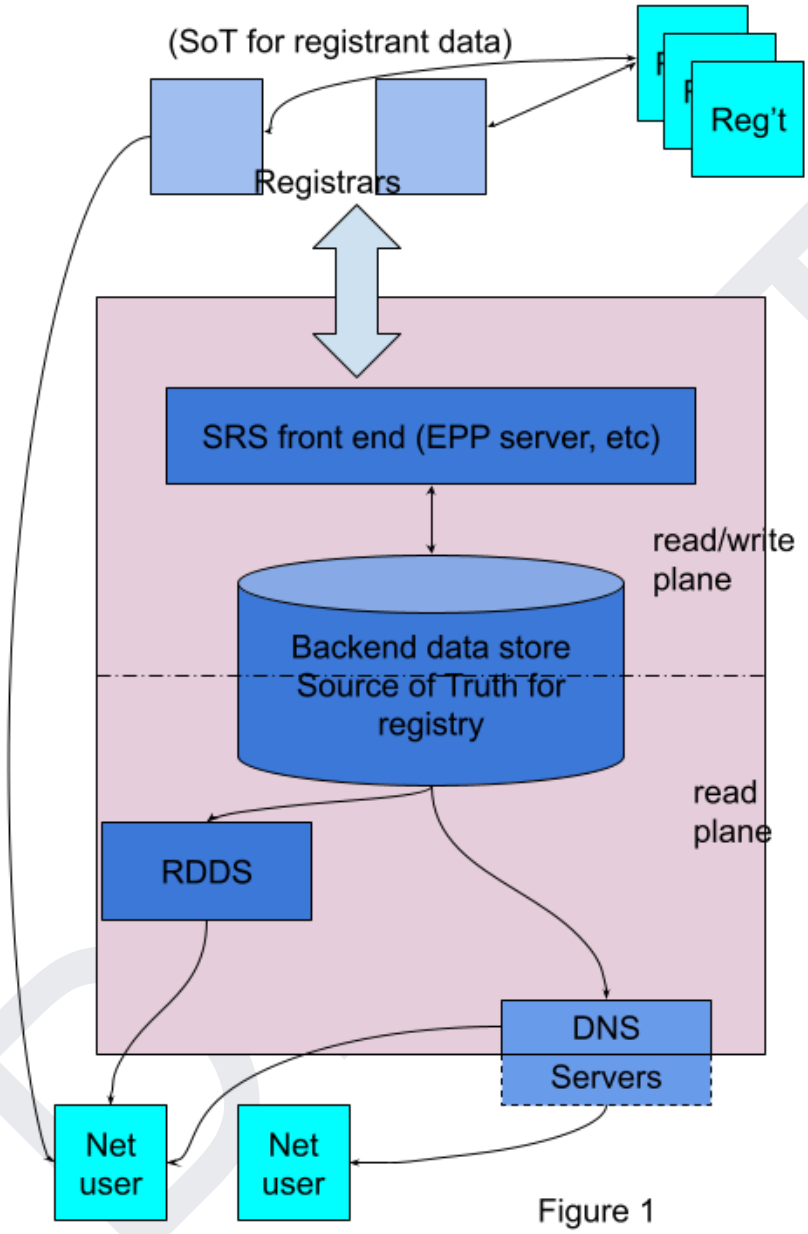
(SoT for registrant data)
Reg't
Registrars
SRS front end (EPP server, etc)
read/write
plane
Backend data store
Source of Truth for
registry
read
plane
RDDS
DNS
Servers
Net
user
Net
user

Figure 1

Contrast that illustration with Figure 2, illustrating a more distributed architecture – one which is possible in the case of integrated global DNS and alternative naming systems. Note that integration on the string+controller model does not *require* this sort of distributed design. In fact, the design could be extremely similar to that outlined in Figure 1. But for this report's purposes, it is useful to imagine a highly decentralized system with a wide degree of distributed operation, and to try to describe the potential for safety and security issues arising from that.

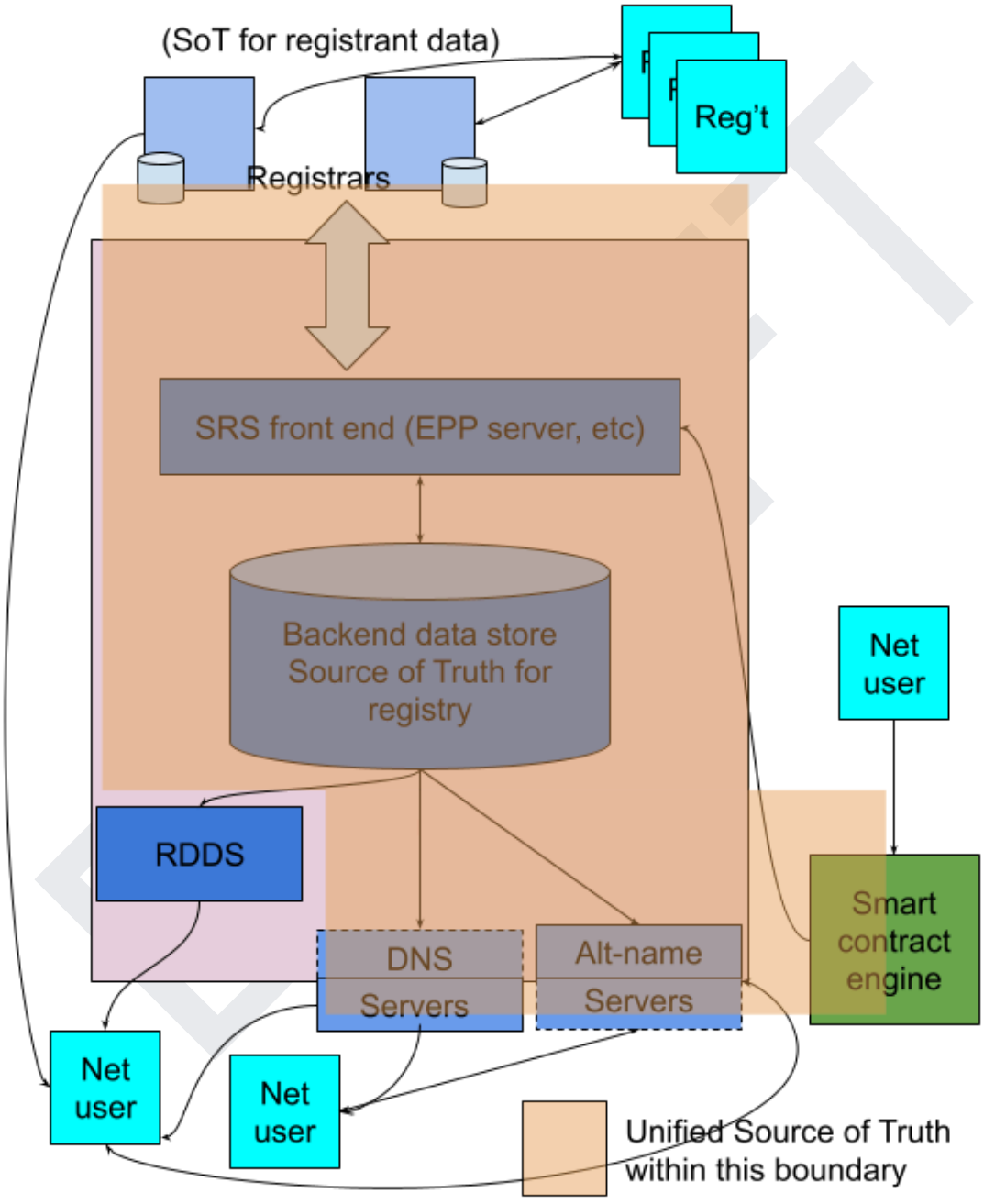

Figure 2

Figure 2 adds elements related to a smart contracts system along with an alt-naming system alongside the global DNS system. The key addition is not those components, but the beige region that outlines how to understand the extent of the Unified Source of Truth. The USoT includes not only the central backend database of the SRS/registry, but also the different naming systems servers, the engine that drives the smart contract system, and the database containing registrant data that exists at the registrars (and not, in this scenario, in the "thin" SRS database). In order to make sure that the USoT contains all the evidence needed for string+controller integration, most or all of these systems need to be consulted *even if they are each under the control of different parties*. The total system may use some kind of optimistic locking, possibly along with multiple-phase commits, and a robust system of pending status values for domain name objects in order to be sure both that a write into the database is not accepted until the state of each relevant component supports it, but also that it check for unchanged data as it writes the changes in the system (and fail or undo write activity if any such change is detected).

This arrangement is, to be sure, considerably more complex than the prior model, and it is, to be fair, a more complicated set of subsystems than seems likely to be used by anyone. Nevertheless, it is not too hard to see how this set of arrangements might work. The registrar-operated RDDS provides some kind of durable identifier (ideally, a cryptographically-protected identifier) that is to be used in the SRS backend to store the identified registrant associated with the allocated and activated names in the SRS database. This is to be true no matter which path any of the transactions that were sent into the SRS back end take. Each such path will eventually deliver the required information to the main data store, which in turn ensures that once every naming system has been appropriately allocated or activated, the naming system servers are populated with the data that is a push from the centre. And, of course, there is always the possibility of depending upon contractual terms for keeping registries and registrars in sync.

7. Considerations for When Integrated Registry Ceases Following Requirements

For the purposes of this report, it is assumed that registries asking to create a string+controller integration will have acted in good faith, and that there is no question as to whether a registry operator planned to meet its obligations under the specifications outlined above. Evidence of bad faith – whether the bad faith was actually exhibited during the original indicated desire to operate an integrated namespace, or during a run-up to financial crisis when concerns about the viability of the registry were not communicated to ICANN or the user community – should all be treated as, fundamentally, the same as any lack of transparency in the TLD's operation and therefore as a compliance issue.

Sometimes, however, despite the best of intentions, an operational launch reveals something in the plan that does not survive contact with the world. In the event of an operating

failure of this sort, the principal resolution of the issues is still likely to be built around contracts and compliance. There are, however, some technical considerations that can be brought to bear.

To begin with, it is not possible to reduce the technical requirements for string+controller integration below those outlined in this report. Proposals to make the integration commercially viable by weakening any of these provisions must be rejected as proposals to substitute some other kind of registry service for string+controller integration. Such proposals would need their own registry service evaluation.

At the same time, there is a possibility that more than one implementation would meet the technical requirements laid out in this report. Under the kinds of operations possible under Enrollment Via Registration in Shared Registration System, it is certainly possible for a registry operator to subcontract certain operations, in exactly the way ROs subcontract component operation of some components in a single-naming-system (DNS) registry. The yet more-flexible arrangements under the description in String+controller Integration in a Distributed Mode explicitly permit different components to be under the management of different parties. This may mean that an operator who struggles to provide a given piece of functionality itself at a cost it can afford may find someone else who has a more efficient model (or greater economies of scale) such that the feature can be maintained in a way that is not, at least, loss-making.

Nevertheless, it would appear prudent to have, as part of the definition of the registry service, a statement of what capabilities the service enables, and a plan for how such capabilities might be turned off in the event the integration is not viable. Until there is a substantial body of experience with integrations of the global DNS and other naming systems, a mandatory “turn-down plan” for discontinuing the integration is RECOMMENDED as part of the evaluation of the registry service.

7.1. Considerations for EBERO

The Emergency Back-End Registry Operator (EBERO) is a mechanism that is invoked when a registry is experiencing enough difficulty that it risks failing to sustain any of five critical registry functions. The EBERO specification excludes services beyond an enumerated list from the scope of emergency operation. Integration with an alternative naming system appears to fall outside the category of “critical registry functions,” which suggests that integration cannot survive EBERO operation. This almost certainly requires a plan for how to turn down integration, which is part of the reasoning to require one as part of the registry service application.

8. Registration, Allocation, and Other Operational Scenarios

This section outlines some common use cases in order to illustrate how the total collection of systems, including the alternative naming systems, work together. The outlines show that the collection of systems work the same way regardless of the originating intent of the scenario. This is a positive sign that the requirements deliver the necessary stability.

8.1. DNS Registrant Wishes to Use example.org DNS-name, No Other Names

- Service checks for availability in each naming system. If no, reject. If yes, proceed → *name enrolled*.[8]
- The registrant offered the choice to allocate DNS-name and any alt-names.
- The registrant chooses only the DNS-name and communicates this to the registrar.
- The registrar performs registration action in SRS for global DNS registry according to historical patterns → *name allocated in global DNS; name withheld in other naming systems*.
- Registrar adds registration elements for activation. *Note: this may not actually be a separate action, but is probably completed as part of the initial registration in the SRS. It is called out here as a separate act in order to highlight the logical state change. → name active in DNS; name remains withheld in other systems.*
- Name operates.

8.2. Registrant Through Traditional Registrar Wishes to Use example.org alt-name, No Other Names

- Service checks for availability in each naming system. If no, reject. If yes, proceed → *name enrolled*.
- The registrant offered the choice to allocate global DNS-name and all alt-names.
- The Registrant chooses a single alt-name, no others, and communicates this to the registrar.
- Registrar registers alt-name via SRS → *alt-name allocated in relevant alt-naming system; name withheld in other naming systems*.
- Alt-name mechanism for activation invoked → *alt-name active in relevant alt-naming system; name remains withheld in other systems*.
- Alt-name operates.

This case just inverts the previous one: the desired outcome from the registrant is to operate *only* an alt-name and not the DNS-name under a traditional registry-registrar operational

[8] As noted when the term enrollment was defined, this step is actually a logical function, and in most systems is provided implicitly through other actions. The point of making the distinction is to see the logical function and how it is separate form fillint in d

model. This is included for completeness and to illustrate how the case works using exactly the same mechanism as the prior case, but it seems an unlikely scenario under the use cases often described.

8.3. Registrant Wishes to Use example.org as DNS-name and at Least One alt-name

- Service checks for availability in each naming system. If no, reject. If yes, proceed → *name enrolled*. Note that in this case during creation more than one system is intended for enrollment, but the check is the same because the availability must always be in every system.
- The registrant is offered the choice to allocate DNS-name and all alt-names.
- The registrant chooses at least one alt-name, and the DNS-name, and communicates this to the registrar.
- Registrar issues domain create command via DNS SRS, and the necessary create command for the alt-name.
    - Could be a single SRS command, depending on implementation.
    - Depending on the mechanism for creation in the alt-name, which could take some time to settle, SRS creation command results in a pendingCreate status (or other similar status).
    - Assuming this settles → *DNS-name and alt-name in desired system allocated; alt-name withheld in other naming systems*. If it does not settle, there is a bug at enrollment, above.

All use cases boil down to one of the three outlined above (which are all really just variations on the basic case, with differences really only in how many and which of the different naming systems are involved). There are some important corollaries. Each case starts with enrollment. This implies that, if an existing alternative naming system is brought into the global DNS registry world with a number of existing alt-names, those names *must be enrolled* and therefore *must be at least withheld from the global DNS*. This is a technical implication of the requirements that may have policy implications needing to be discussed with the community but that are beyond the scope of this report. Second, any case where the alt-name is manipulated directly by an outlying system (e.g., a name automatically assigned as part of a blockchain transaction) still entails updates including all the criteria outlined in this report.

8.4. Registrant Uses example.org as a DNS-name and alt-name; DNS-name is Permitted to Expire

- On expiration of DNS entry, DNS-name is withheld from the global DNS.
- What happens with the alt-name depends on two things:

    - What is the expiry/renewal policy of the alternative naming system? Do names even expire in that system?
    - Does the registrant want to maintain the alt-name and not use the DNS-name?
- If the alt-name is to be maintained, then the DNS-name remains withheld and the alt-name remains active.
- If the alt-name is not to remain active, then all the names in all the systems are disabled (that is, the name is deactivated), and then de-allocated, and finally removed from enrollment.
- After that, the name is available.

8.5. Example.org Comes Under Some Kind of Administrative Deactivation in any naming system

- When names are deactivated in the global DNS (sometimes using the clientHold status, which is imposed by the registrar; or serverHold status, which is imposed by the registry), it is usually due to some kind of serious issue with the domain name. Often, the names are taken out of service based on security or stability grounds for the Internet.
- The nature of such *Hold status values depend on the EPP extensions created to permit integration. It will be necessary that holds are possible in one naming system and not another.
- If a service is deactivated for external reasons (e.g. abuse, court order) in one naming system, the registry MUST disable the name, on the prudential principle that a danger to the Internet from a given name in one naming system is a danger in all naming systems (particularly if the naming systems are integrated).
- A clientHold status (from the registrar) SHOULD NOT entail the disabling of a name, since it can be used on purpose only to stop global DNS resolution on a domain for some other reason. This is an example of an area where EPP extensions seem likely to be necessary to make management of integrated names easier. If a registrar is adding a clientHold to undertake disabling a name, it SHOULD place the clientHold for the same name in every naming system.

8.6. Example.org Comes Under Suspension In Any Naming System

- Suspension does not remove DNS-names from resolution on the Internet, but prevents administrative actions on the name.
- Suspensions MUST apply to all naming systems simultaneously.

  Note: perhaps unintuitively, suspensions due to policies like Uniform Rapid Suspension do not remove the name from operation in the global DNS. That is the meaning of “suspension” used here.

8.7. Example.org Undergoes an “Internal Transfer” (i.e., a Change of Registrant) as a DNS-name

There are four possible cases for this example:

1. In the first case, the name is entirely managed in the SRS, the alt-name and DNS-name are properties of the domain object, and the registry has a registrant property or object somehow associated with the domain object. In this case, there is an EPP extension to the Domain Object Mapping and there is not a separate object mapping. This case guarantees that the string (which is the <domain:name> element of the EPP object) is the same for both the global DNS and alternative naming system(s). Because there is a registrant associated with the domain object, it also guarantees that the controller is also always the same, and so the basis of integration is guaranteed. This case is illustrated in a flowchart in Diagram 3.

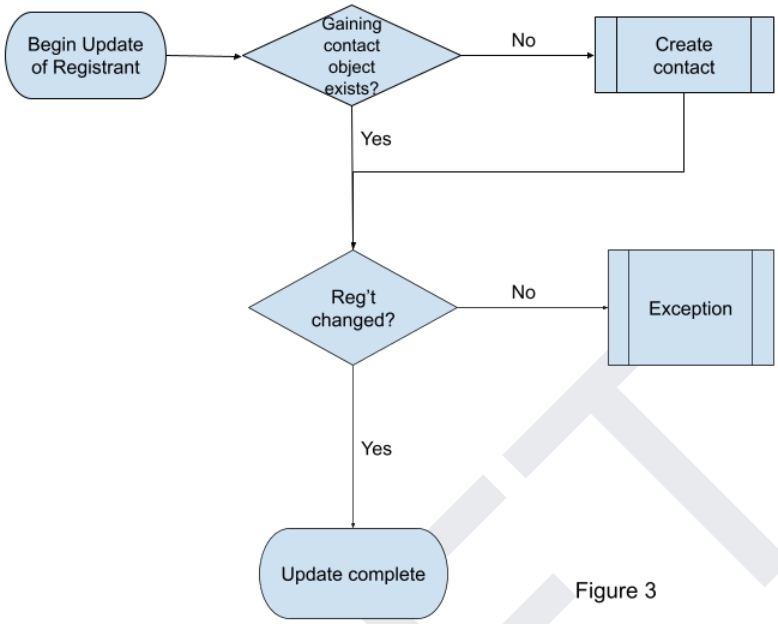

Figure 3

2. The second case is something like the first, only the registry does not have the registrant information associated with the domain object. Instead, the registrar has that data (so the flowchart is still in Diagram 3). From a technical point of view, the challenge here is the assertion that the integration of the different naming systems by string+controller is a *registry* service, since the registry is not technically in a position to assert all the conditions of integration at all times. And indeed, in the case of a registrant change under this model (a so-called "internal transfer"), everything depends on contractual compliance. It is entirely possible that the registry never learns that the change has happened, since the change can be effectuated completely within the business practices of the registrar without affecting the registry at all.[9] Nevertheless, if it can be demonstrated that the registrar's domain registration is always a single object with a single registrant, then this second case is practically like the first in effect, only using different evidence to prove it. If the registrar, however, uses different objects or different contact information for the DNS-name and the alt-name, then the string+controller integration is not guaranteed, and must be demonstrated in some other way. There is no illustration for this case provided here because there are too many possible combinations of ways to achieve it. All of them require consultation between the registry and the registrar if the registry is understood as providing the service.

[9] Of course, cases like this might be handled as a kind of hybrid, in which the registry receives a minimal amount of information about the registrant but has enough to track its persistence of identity through time. In fact, however, any such case can be reduced to the first one, since as long as there is some reliable and persistent link between the string containing the name and a reliable identifier of the controller stored in the SRS, that is enough to assert string+controller integration.

3. The third case is when the name is entirely managed in the SRS, but there is a different object mapping than for the domain object. This case holds similarities to the problem of subordinate host object and domain object mapping in EPP registries, for those familiar with that issue. A domain name object cannot be renamed under EPP, and this report assumes that an EPP object mapping for an alt-name would make the object name similarly immutable. Therefore, the link between a domain object and an alt-name object would ensure that the string identity would be maintained. However, because the two objects could be associated with different controllers (either within the EPP repository operating as the registry, or else within the client systems operated by the registrar) it would be challenging to guarantee the string+controller properties under all circumstances. An obvious way to solve this is a registry-based stored procedure that transfers the two objects as the result of a single command. That is illustrated in Figure 4.

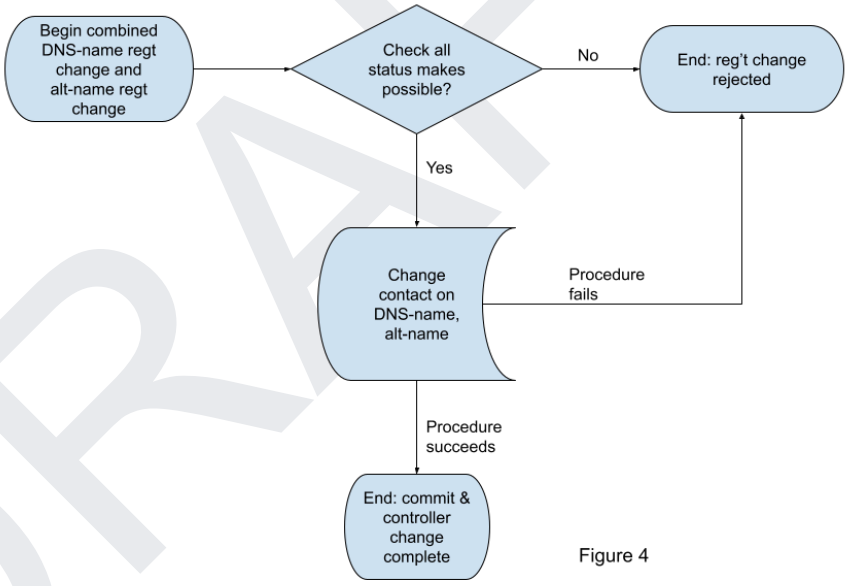

Figure 4

4. The fourth case is, in effect, any case where the DNS-name and the alt-name are not managed through the same SRS. This, like any other USoT example, is too generic to allow for a single analysis of the operational scenario. In theory, however, the USoT is supposed to ensure that the string+controller condition is never violated. Depending on the guarantees offered by the USoT, this assurance may be easier or harder to demonstrate. It would be one of the most important demonstrations an applicant for a registry service would need to offer. A flowchart to illustrate this case is not practical because the case is too generic.

8.8. Example.org Experiences the Domain Transfer Operation (i.e., a Change of Registrar) as a DNS-name

Domain transfers are among the more complicated operations in an SRS, and therefore appear to provide a good exercise of how the global DNS and alternative naming systems together either meet the requirements or have gaps in how they meet such requirements. The discussion here is illustrative, not exhaustive, but the general cases are the same in any event: either the registry does or does not *itself* have all the data it needs to complete the domain transfer.

- The name example.org is enrolled, allocated in all naming systems, and active in all naming systems.
- An SRS domain transfer command from a gaining registrar is received. The basic workflow is like any normal domain transfer: if the domain status permits such a transfer and the appropriate conditions are met, the name enters pendingTransfer; otherwise the transfer fails. This is illustrated in Figure 5.

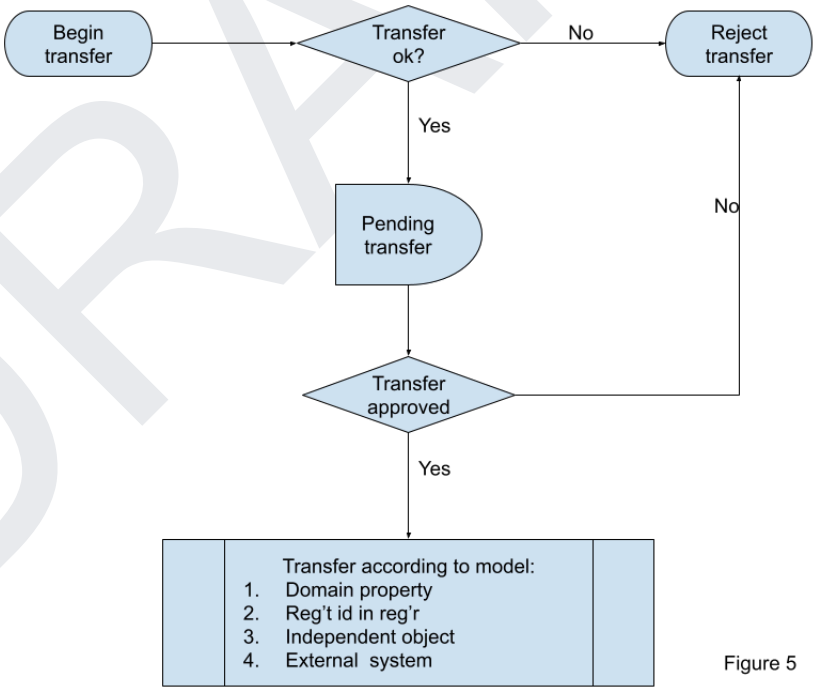

Figure 5

- The simplest case is a straightforward one where the alt-name is associated with the domain object via properties, and the registrant is associated with the domain object. In this case, the transfer (after the

pending period or if approved by the losing registrar) is just an update of the registrar association, illustrated in Figure 6.

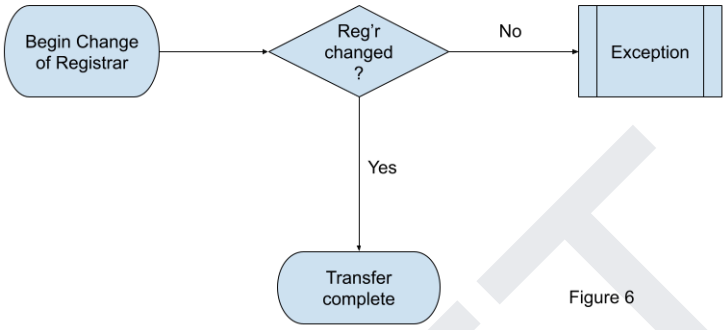

Figure 6

- As in the case of updating the registrant (see previously), if the registrant information is actually only at the registrar, the matter becomes complicated. It can be illustrated as in Figure 7, but the diagram is misleading. The “preparation” hexagon does not illustrate that the answer to the question comes from the gaining registrar, not the registry, and could *in principle* change while the registry is working. That the gaining registrar is in some kind of relationship with the domain is what authInfo is for: it is a bearer token. The risk is almost certainly very small, but there are timing-based attacks and race conditions that need to be assessed and, if spotted, addressed.

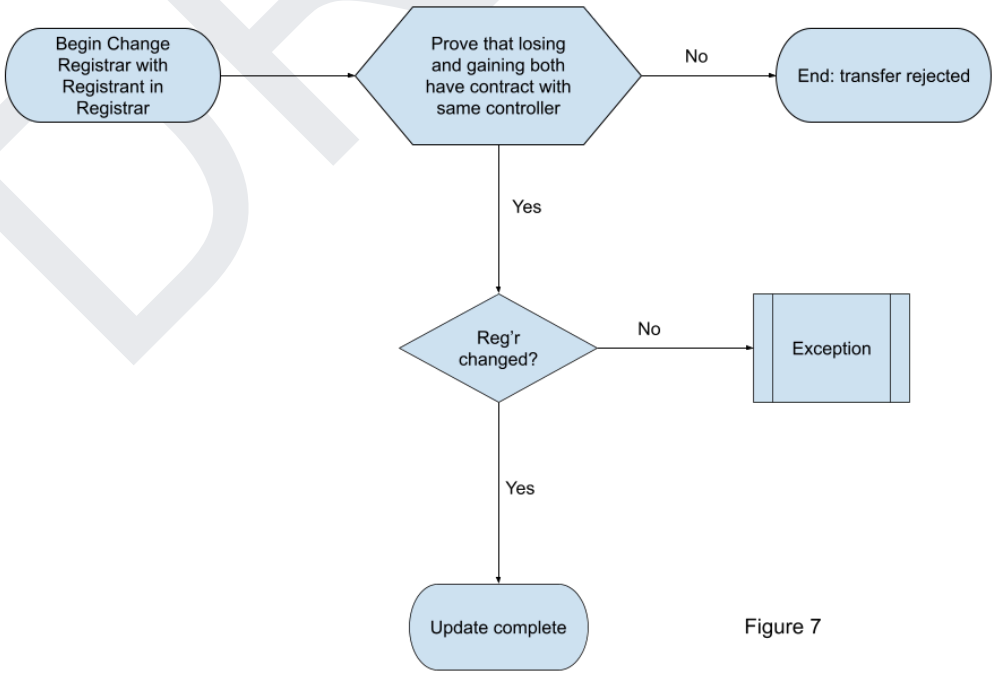

Figure 7

- If the domain object (for the DNS-name) is separate from the alt-name object, then the workflow is substantially similar to what happens in the case of changing the registrant, and is shown in Figure 8.

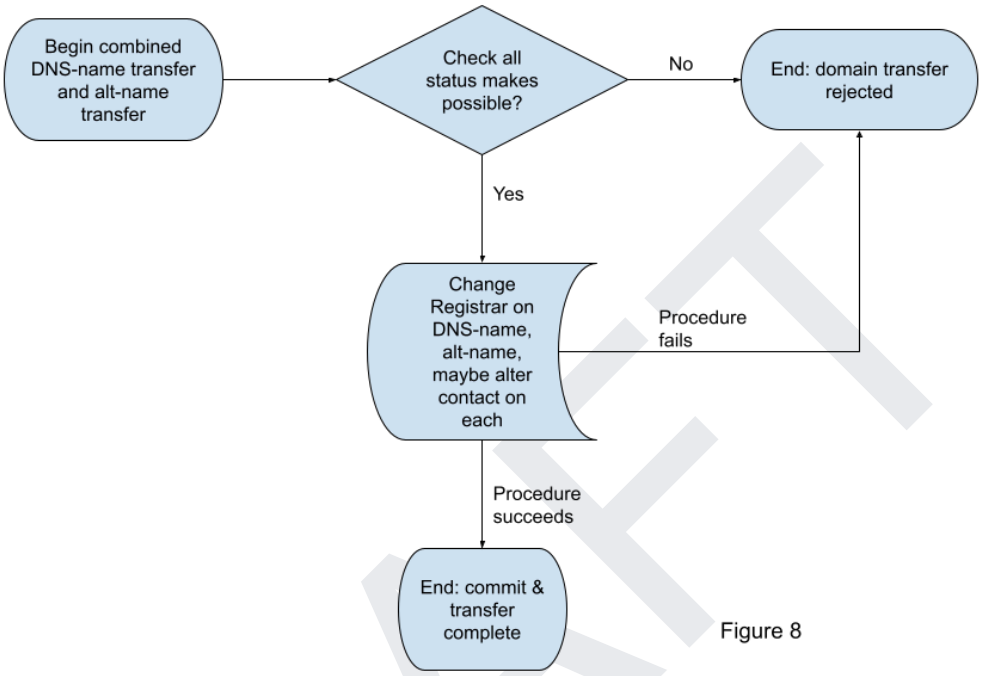

Figure 8

- Finally, as in the case of changing the registrant, if the alt-name or substantial information about it is stored in systems outside the registry, maintaining the string+controller condition is complicated, although not dramatically more complicated for a domain transfer than for anything else in the USoT mode of operation. This case cannot be illustrated usefully because there are too many possible ways for the external systems to work to be represented in a single diagram.

9. Technical and Practical Feasibility

It is both technically and practically possible to implement a system as outlined in the sections above so that a DNS-name and one or more alt-name(s) are always maintained such that they use the same string (i.e., that they are the same name, and that the name is immutable); and that they are maintained by the same controller. It is also clear that not every possible thing someone might want to do with alternative naming systems can be supported within the requirements laid out in this report. Nevertheless, there are several modes of operation that would be possible to implement under these requirements. Following are sketches of a few such modes as evidence, but the sketches are not comprehensive; neither

does this cover all the possible things one might do within the requirements covered by this report.

9.1. DNS Registry Is Primary

One approach to integration is to require any data-changing activity to flow through the traditional Shared Registration System (SRS) of the existing DNS registry. This approach treats the alternative naming system as another system where publication of registration data can happen, just like the global DNS and RDDS. Since every transaction changing the data passes through the SRS, any downstream system is subject to the same control. In principle, then, the consistency of operation of the name itself can be demonstrated, even if the domain object (for the DNS-name) and an object for the alt-name are not the same object. This still, however, does not prove that string+controller integration is maintained.

This arrangement only provides technical proof that the integration is maintained in case the registry contains an object showing that the controller for the DNS-name and the alt-name are the same controller. This might be achieved by using a registrant contact object that is kept the same for the DNS-name and alt-name. The contact object might not contain full contact details for any party, so long as it contains enough such detail to show that the controller is the *same* controller in the case of the DNS-name and any alt-name of the same string. The way such detail might be indicated is one of the necessary details a prospective registry service operator should include in their application materials. This sort of mechanism is contemplated in the discussion in Enrollment Via Registration in Shared Registration System. Alternatively, the assurances of the controller identity may be made through contractual terms between registrars and the registry, as noted in String+controller Integration in a Distributed Mode.

Note that a single SRS may not ensure that proof of string+controller can always be found in a single database. Many global DNS registries are “thin,” and do not have contact information for a given domain name registrant. From a technical perspective, that data would need to be queried from the appropriate registrar. That kind of arrangement is more along a spectrum towards the “emergent truth” of the USoT than is deriving the true state of the world from a single database.

This approach to integration requires the specification of changes to the protocol used to manage the SRS. In the case of ICANN gTLDs, this is EPP. There is more than one way EPP could be changed to meet the need, and the history of EPP suggests more than one way will be tried. A registry planning to undertake integration using the registry interface will need a means to create and manage an alt-name. This could be in relation to a domain object, or via a separate object mapping. The challenge implicit in using a domain object for this is that the domain object in EPP is principally designed to support DNS-names in the first place, and may be a poor match for data arising from an alt-name. The challenge implicit in using an object mapping is rather like the challenges that EPP registries have faced in using shared objects like “internal hosts” in EPP repositories. In the case of alt-names, that problem might be more acute, in that the alt-name would not be a subordinate object to the DNS-name, but would essentially be another object in the EPP repository with the same priority as the domain object. Regardless of how one proceeds, an alt-name seems an awkward fit for EPP.

Much less awkward is the use of contacts in the registry to track the controller that is the second part of string+controller integration. Unfortunately, however, many registries appear to be reducing the amount of data they store and moving to a “thin” model. Any prospective registry service provider will need to specify how their service will track the controller of integrated domains, especially if they have no plans to require contact information within the registry.

As noted in Enrollment Via Registration in Shared Registration System, this style of operation automatically supports all the existing ICANN consensus policies and contractual requirements to whatever extent an operator’s DNS-name registry operation and contract supports all of that. In that sense, the prospective registry service provider’s task seems somewhat easier in this case, since many security and stability questions that might arise in the preliminary evaluation are answered by this mode of operation.

9.2. Alt-naming System Is Primary

An alternative approach is to invert the previous integration, and to use the (or one of the) alternative naming system(s) to control activity flowing through the collective systems. While this arrangement may be less clear as to the center of control compared to the description in the previous approach, that may be only due to familiarity. The same basic structure for transaction control would prevail, but the principal control would be something other than a global DNS registry SRS. Changes would flow from whatever the change-control system is in place for the alternative naming system through other systems.

This approach to integration has many of the advantages and disadvantages of the prior one, with the possibly fatal disadvantage for a registry service provider that it does not build on any of the familiar infrastructure of the registry-registrar-registrant model and yet still needs to exercise approximately all of it (by virtue of using the global DNS as one of the integrated names). For that reason, this approach seems like a difficult way to provide a registry service.

9.3. Bearer Token Proves Unity

An entirely different approach to centralizing operations through a single point of control is to operate in a highly decentralized manner. This approach might be implemented through cryptographic tokens that prove control: an appropriately-derived token might appear in the global DNS at the DNS-name, and in the alternative naming system(s) at the alt-name, using techniques along the lines described in “Domain Control Validation using DNS” (Sahib et al., n.d.). By using some common seed as a challenge mechanism to a prospective change to the data about any of the affected naming systems, it is possible for a registrant to demonstrate control over the integrated name in all of the possibly affected systems. This arrangement is far from the way most gTLD registration systems have historically worked, but it is in line with mechanisms widely used to establish control over a domain name before delivering Internet services to the purported operator of that domain name.

A significant challenge for this mode of operation is to demonstrate that it can meet all ICANN contractual requirements and consensus policies for those names that are integrated

with the global DNS (because the TLD is integrated) but where those names themselves are not part of the global DNS and therefore have not provided required contact data and so forth for the name. If example.net is operating only the alt-name and is not operating the related global DNS-name, and has not provided registrant data for example.net to anyone, it is not immediately obvious how policies like Uniform Rapid Suspension might work or how the Uniform Dispute Resolution Policy can operate in the absence of standard contact data. The definition of “stability” in the RESP, especially, leaves room for including the legal and social dimensions of Internet operation as part of the stability that needs to be reviewed for a policy to be acceptable. This challenge need not be insurmountable, but it does appear to be an issue that would need to be faced by any prospective *registry* service provider seeking to integrate a TLD with alternative naming systems. If the burden of compliance in this respect appears too high, the TLD in question might not be a good candidate for string+controller integration. On the other hand, more than one alternative naming system operating on the Internet today may well be a better poised to use the techniques of distributed operation as the path to integration, and if the registry services model is too onerous to make integration work there may be a significant “shadow operation” of alternative naming systems that emerges. That shadow might correspond with the DNS-names officially sanctioned by ICANN, and it would be unfortunate if that shadow emerged in part because alternative naming system operators found the ICANN/IANA root procedures too burdensome to be worth integrating with.

10. Additional Security and Stability Considerations

It’s difficult or impossible to characterize the full range of security and stability considerations covering global DNS, another naming system, and the mechanisms by which both are generated from a shared source or sources of relevant data. However, there are a few principles and some available guidelines that operators should be aware of. This report attempts to lay out a number of fundamental ways.

There is a large body of knowledge and best practices available for global DNS, and operators of integrated namespaces should use them for guidance in providing a secure environment for global DNS, the integrated naming system, and the shared source of registration data. In particular, the DNS integration Internet-Draft discussed previously (Sheth et al., n.d.) includes a list of factors to be considered, such as the stages of the domain name lifecycle and the security and reliability of synchronization mechanisms. These are contained in Section 3 of that draft.

In addition, it is expected that operators of gTLDs delegated from the root of the global DNS will have Registry Agreements (RAs) with ICANN. The base RA, and specific additional conditions as may be negotiated between ICANN and the registry operator, contain a number of requirements for the security and stability of global DNS and SRS operations. For example, the current base RA provides that the DNS gTLD zone must be signed with Domain Name

System Security Extensions (DNSSEC), and that the operator must be able to show a "DNSSEC Practices" document describing such features as key management. While these requirements are DNS-specific to date, at least some of them will also apply to an integrated naming system. The following is not an exhaustive list, but illustrates some of the issues that a prospective registry service operator wanting to operate a DNS-name integrated with at least one alt-name will need to ensure are addressed:

- What assurances of data integrity from lookup data do people get? In other words, why isn't DNSSEC needed for this alt-name?
- Do consumer expectation protections (such as URS or UDRP) currently work in the alternative system? If so, how? If not, how will they work under integration?
- What anti-abuse regime covers alt-names? Is it enough?
- Whom can one contact in the event a system becomes the source of malicious traffic on the Internet?
- How long has the alternative naming system been operating, under what loads, and how well has it been vetted by both its standards development organization (SDO, assuming it has one) and by operators?
- What potential is there for a network split that leaves some of the naming systems partly isolated. What to do or not do about that?
- What potential is there for a legal authority to break the object+controller integration, thereby possibly compromising security and stability in an effort to achieve some (other) social good.

10.1. Name Hierarchy and Naming Systems

There is no requirement that integration persist at every level of the global DNS hierarchy even if a given name is integrated with another naming system and requires or permits integration at immediately-subordinate names. To state this another way, there is no expectation that integration continues all the way down the tree. So a registrant of example.net might use integration across multiple naming systems, and yet MAY use internal.example.net in completely unrelated ways in the global DNS and in other naming systems, even delegating that name to different external parties (though such an arrangement might be unwise). At the same time, it is RECOMMENDED that a public suffix (Mozilla Foundation, n.d.) either maintain the integration lower in the tree or else disable integration for the domains in question. Under the mechanism described in this document, disabling such integration is always a possibility for the registrant of any name delegated in the global DNS. Since no one could be the registrant of a domain name subject to integration and *not* have control of the same name in the alternative naming system in question, no one else could use that domain name in the alternative system. If the operator of the DNS-name is not in control of the alt-name, then the name is simply not a candidate for integration in the first place.

10.2. Interaction With IDN Policies

It is possible that the integration of global DNS and other naming systems will happen in a registry with complex Internationalized Domain Name (IDN) policies, including policies for variants and significant development of the Label Generation Rules (LGR) for character repertoires of different writing systems. In addition, alternative naming systems could use different normalization forms than IDNA (Klensin 2010). Any string+controller depends on using the *same* string, however, and therefore any variant generation or LGR processing MUST occur prior to integration of the naming systems. It is possible that, in some implementations, the integration needs an integration of the variant or LGR elements of IDNA processing in order that the rules for characters are handled correctly under different normalizations. There are good reasons to question the wisdom of operating using multiple normalizations simultaneously.

11. Items Not In Scope For This Report

1. Mere use of names within alternative systems, i.e., if the name is not used as an *identifier* in the alternative system, then this report has not considered that case. If it is not possible to query the alternative naming system for a name, then it is not a case covered in this report.
2. Long-term viability or utility of non-DNS naming systems for general Internet-wide use. Whether any particular business model integrating the DNS and other naming systems will work in the long term is not part of any analysis undertaken by the study group.
3. Policy and competition matters in general, and appropriateness of any particular ICANN policy to a given naming system in particular.

12. Personnel and Acknowledgments

The TSG members were assembled from the community, and no member represented a particular constituency. Their affiliation is provided below for identification purposes. Members served as peers.

Name	Affiliation
Sebastien Ducos	Unstoppable Domains
Nick Johnson	ENS Domains
Brian Lonergan	Identity Digital

Name	Affiliation
Georgia Osborn	Security and Stability Advisory Committee
Don Ruiz	Orange Domains
Swapneel Sheth	Verisign
Peter Thomassen	deSEC
Suzanne Woolf	Public Interest Registry

ICANN is grateful to the members for their generous donation of their time and expertise. The TSG was supported by ICANN staff and contractors.

13. References

Blanchet, M. 2022. *Finding the Authoritative Registration Data Access Protocol (RDAP) Service*. No. RFC9224. RFC Editor. https://doi.org/10.17487/RFC9224.

Bradner, S. 1997. *Key Words for Use in RFCs to Indicate Requirement Levels*. RFC 2119. Internet Requests for Comment. RFC Editor. https://doi.org/10.17487/RFC2119.

Durand, Alain. 2022. *Challenges with Alternative Name Systems*. OCTO No. 034. Internet Corporation for Assigned Names and Numbers. https://www.icann.org/en/system/files/files/octo-034-27apr22-en.pdf.

Galvin, J., and M. Bauland. n.d. “Same Entity Set Support for EPP.” Work In Progress No. 05. Internet-Draft. Https://datatracker.ietf.org/doc/html/draft-galvin-regext-epp-variants. Accessed August 9, 2026. https://datatracker.ietf.org/doc/html/draft-galvin-regext-epp-variants-05.

Hoffman, P. 2024a. *Introduction to Blockchain Name System Technologies*. OCTO No. 039. Internet Corporation for Assigned Names and Numbers. https://www.icann.org/en/system/files/files/octo-039-17oct24-en.pdf.

Hoffman, P. 2024b. *Introduction to Blockchain Technologies*. OCTO No. 040. Internet Corporation for Assigned Names and Numbers. https://www.icann.org/en/system/files/files/octo-040-17oct24-en.pdf.

Hoffman, P., and K. Fujiwara. 2024. *DNS Terminology*. Request for Comments RFC 9499. Internet Engineering Task Force. https://doi.org/10.17487/RFC9499.

Hollenbeck, S. 2009a. *Extensible Provisioning Protocol (EPP) Domain Name Mapping*. Request for Comments RFC 5731. Internet Engineering Task Force. https://doi.org/10.17487/RFC5731.

Hollenbeck, S. 2009b. *Extensible Provisioning Protocol (EPP) Host Mapping*. Request for Comments RFC 5732. Internet Engineering Task Force. https://doi.org/10.17487/RFC5732.

Hollenbeck, S. 2009c. *Extensible Provisioning Protocol (EPP)*. Request for Comments RFC 5730. Internet Engineering Task Force. https://doi.org/10.17487/RFC5730.

Hollenbeck, S., and N. Kong. 2015. *Security Services for the Registration Data Access Protocol (RDAP)*. No. RFC7481. RFC Editor. https://doi.org/10.17487/rfc7481.

Hollenbeck, S., and A. Newton. 2021a. *JSON Responses for the Registration Data Access Protocol (RDAP)*. No. RFC9083. RFC Editor. https://doi.org/10.17487/RFC9083.

Hollenbeck, S., and A. Newton. 2021b. *Registration Data Access Protocol (RDAP) Query Format*. No. RFC9082. RFC Editor. https://doi.org/10.17487/RFC9082.

ICANN. n.d.-a. "Bylaws FOR INTERNET CORPORATION FOR ASSIGNED NAMES AND NUMBERS | A California Nonprofit Public-Benefit Corporation." ICANN, Internet Corporation for Assigned Names and Numbers. Accessed August 9, 2026. https://www.icann.org/en/governance/bylaws.

ICANN. n.d.-b. "Registry Services Evaluation Policy." ICANN, Internet Corporation for Assigned Names and Numbers. Accessed August 9, 2026. https://www.icann.org/en/contracted-parties/consensus-policies/registry-services-evaluation-policy.

ICANN IDN Variant Issues Project. 2012. *A Study of Issues Related to the Management of IDN Variant TLDs (Integrated Issues Report)*. Internet Corporation for Assigned Names and Numbers. https://www.icann.org/en/system/files/files/idn-vip-integrated-issues-final-clean-20feb12-en.pdf.

Kaizer, Andrew, Will Naciri, and Swapneel Sheth. 2024. "Poster: Synchronization Concerns of DNS Integrations." *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, December 2, 4982–84. https://doi.org/10.1145/3658644.3691415.

Kaizer, Andrew, William Naciri, and Swapneel Sheth. 2025. "Synchronization Concerns of DNS Integrations." Preprint, July 26. https://doi.org/10.36227/techrxiv.175355215.52122651/v1.

Klensin, J. 2010. *Internationalized Domain Names for Applications (IDNA): Definitions and Document Framework*. Request for Comments RFC 5890. Internet Engineering Task Force. https://doi.org/10.17487/RFC5890.

Leiba, B. 2017. *Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words*. RFC 8174. Internet Requests for Comment. RFC Editor. https://doi.org/10.17487/RFC8174.

Mockapetris, P. V. 1987a. *Domain Names - Concepts and Facilities*. No. RFC1034. RFC Editor. https://doi.org/10.17487/rfc1034.

Mockapetris, P. V. 1987b. *Domain Names - Implementation and Specification*. No. RFC1035. RFC Editor. https://doi.org/10.17487/rfc1035.

Mozilla Foundation. n.d. “Public Suffix List.” Accessed August 9, 2026. https://publicsuffix.org/.

Newton, A., B. Ellacott, and N. Kong. 2015a. *HTTP Usage in the Registration Data Access Protocol (RDAP)*. No. RFC7480. RFC Editor. https://doi.org/10.17487/rfc7480.

Newton, A., B. Ellacott, and N. Kong. 2015b. *HTTP Usage in the Registration Data Access Protocol (RDAP)*. No. RFC7480. RFC Editor. https://doi.org/10.17487/rfc7480.

Sahib, S., S. Huque, P. Wouters, E. Nygren, and T. Wicinski. n.d. “Domain Control Validation Using DNS.” Work In Progress No. 13. Internet-Draft. Https://datatracker.ietf.org/doc/draft-ietf-dnsop-domain-verification-techniques/. Accessed August 9, 2026. https://datatracker.ietf.org/doc/draft-ietf-dnsop-domain-verification-techniques/.

Sheth, S., A. Kaizer, B. Newbold, and N. Johnson. n.d. “Integration of DNS Domain Names into Application Environments: Motivations and Considerations.” Work In Progress No. 04. Internet-Draft. Https://datatracker.ietf.org/doc/draft-ietf-dnsop-integration/. Accessed August 9, 2026. https://datatracker.ietf.org/doc/draft-ietf-dnsop-integration/.

The Unicode Consortium. 2025. *The Unicode Standard*. Version 17.0.0. The Unicode Technical Consortium. https://www.unicode.org/versions/Unicode17.0.0/.

Unicode Consortium. 2025. *Unicode Standard Annex #15: Unicode Normalization Forms*. Annex. Version 17.0.0. tr15-58.html eds. https://www.unicode.org/reports/tr15/.

Valapu, Sulyab Thottungal, John Heidemann, Mattijs Jonker, and Raffaele Sommese. 2026. “Zombies in Alternate Realities: The Afterlife of Domain Names in DNS Integrations.” arXiv:2605.06880. Preprint, arXiv, May 7. https://doi.org/10.48550/arXiv.2605.06880.