

Dear ICANN,

First of all, I would like to thank ICANN for initiating this project (the IDN Variant Issues Project) and also for giving the community the opportunity to comment on the final draft report Jan 2013 titled: “examining the User Experience Implications of Active Variant TLDs”

My intention to provide the following comments is to complement the hard-work provided by the report’s authors in this very important and crucial subject (variants) so that we (all) have a practical and feasible solution that serve the whole internet community at large.

I will start with general comments on the report and then with more specific to its contents.

One major concern that I have is that the report discussions and hence some of its recommendation are OUT OF SCOPE of the project goals and objectives. Regardless of the unquestionable decent intention and the goal to cover other areas beyond TLD level with some nonbinding recommendations they are, nevertheless, considered out of scope.

Another observation is that the report and its recommendations diminish the language community contributions and authorities over IDN issues while they are closer to the user’s needs. But, on the contrary, it expands and broadens ICANN responsibilities beyond its mandate and roles while ICANN is far away from user’s needs and requirements.

General Comments:

- Even though the scope of the report is limited to the usability study of those IDN TLDs that have variants, the report contains discussions and recommendations outside that scope. Thus, the report must limit its scope solely to its mandate which is usability study of those IDN TLDs that have variants. It should not go beyond that limit as it would have misleading conclusions on areas not properly covered in the study. Hence, the report should limit its scope and recommendation to IDN TLDs.
- The report gets involved in restricted area that is considered sovereign to nations. As the report should have concentrated solely in the IDN TLD level, it includes some policies and recommendations to be imposed on countries (registries) which they have their sovereignty over their ccTLDs.

- While the scope of the report is solely for IDN TLDs level which might have conservative approach at the TLD level, its must not be valid for other levels of domain name labels.
- The existing DNS system (without variants) provides some freedom to registries in defining many directions in managing their registries (e.g., database fields, number and type of contacts, number of dns servers, domain name life cycle, prices, renewals, ... etc). This has not deferred registrants and technical community from using and accepting these differences.
- Section 5 listed several challenges related to active variant TLDs. Even though they appear accurate and true from the first look, but they are not caused by variants. Some of the challenges already exist in the current (IDN or ASCII) domain names. For example, if an entity has multiple domain names due to protection against fishing or domain squatting would face some of the same challenges outlined in section 5. Some of the challenges outlined in Section 5, e.g., search engine optimization, is complex and dynamic and is not related to only variants.
- The report (particularly section 5) has succeeded beyond any doubt to frighten and horrify the reader from variants. But it fails to illustrate the problems and security issues if variants are not supported or activated. The need for variants is quite important and crucial. We should not be frighten or consider it as a complex just because it is new or because we are not familiar with.
- It could be understood from the report that the Arab states that got Arabic IDN TLDs have violated or worked against the recommendations outlined in RFC 5564 that was developed and endorsed by the Arab League. This is of course not the case. The RFC 5564 provides the general and basic rules that were developed to cover some linguistic issues that are related to the development of the IDN protocols. It was agreed among the Arab team (who developed the RFC) that local registries could amend it with their local needs.
- In many occasions we recommended determining the accepted characters (code points) for writing an IDN label should be based on languages rather than script table. Many of the confusions and risks outlined in this report can be avoided through adopting language table for each registry. Users write and read a language and not a script.
- It is not necessary for the end user to know or to be able to type all the variants. He/she needs to be able to write one form of the variants (using his/her keyboard) and reach the ultimate resources (e.g., web site). It is the responsibility of the registry to make this happen transparent to the user.

Specific Comments:

Pg 6, Pa 4:

“To learn from existing deployments of IDN variants, this report summarizes and compares how nine ccTLD registries are offering variants for Arabic, Chinese, Devanagari and Latin scripts. Analysis shows that these management practices have many similarities. In particular, although the label generation rules for variants differ across scripts, they all treat the variant labels as a set for all aspects of the domain name life cycle, limit the number of activated variants to ensure a positive user experience and withhold other inactive variants for security and stability purposes (except the case of Latin IDN variants). However, although the Chinese script registries share the same tables and have cohesively defined variants for both the top- and second-level domains, the Arabic script community exhibits some differences within the Arabic language and across languages that share the Arabic script.”

COMMENTS:

- This is not totally accurate. There are no significant differences within the Arabic language communities. All are based on the RFC 5564. Local registries could amend it with their local needs. The differences are reside across languages that share the Arabic script.

Pg 13, Pa 5:

“In addition to differing expectations, there can also be very different implementations of IDNs and their variants, if they are not consistently managed. This can be sub-classified into two aspects. First, what are the various states of a variant (see Section 5 of Integrated Issue Report [4]) and how a variant is assigned to a state (e.g. automatically through a LGR or arbitrarily chosen by the end user, etc.)? Second, how will the variants which are active (if any), be implemented? Differences in how variants are assigned different states, and how the activated variants are implemented would cause confusion in both the use and technical implementation. For example, if no common mechanisms are agreed upon, the activation and resolution of variants may result in different user experiences across”

COMMENTS:

- This is similar to the existing domain name life cycle. Each registry has its own implementation without problem or user confusion.

Pg 14, Pa 2:

“Irrespective of their implementation, the way variant sets are defined and managed could still vary greatly down the tree, e.g. between TLDs and SLDs because TLDs are defined at the script-level and SLDs may be defined at the language level. This will be one major source of variation in the way variants are enabled for users. This change in definition of variants across various levels could become a source of confusion and may have a significant impact on the usability of variants. If user confusion is to be minimized, this variation will need to be managed.”

COMMENTS:

- This is aligned with our recommendation to use the language to determine the accepted characters (code points) for writing an IDN label instead of using the script table. Users write and read a language and not a script. So it is easy to set and implement policies in clear and well-defined.

Pg 14, Pa 3:

“How the variants are enabled and expected to be used by a script community may also vary across script. For example, variants in Chinese script may be visually distinct but those in Arabic script may be visually similar. Some variants may be determined by language and some arbitrarily based on the implementation. For a better user experience, such script level differences should at least be documented, and, whenever possible, minimized.”

COMMENTS:

- Documentation is necessary all the time. Variants are determined by the local communities regardless of their sizes or numbers. They are bound by the need and not limited by the size.

Pg 14, Pa 4:

“The user experience with variants of IDN TLDs may also be very different due to the diversity of technology platforms being used. User systems are configured in a variety of ways, including different operating systems, different keyboards, different fonts, different personalized settings, different applications, different locales, etc. Therefore, even if other sources of variations are contained, the range of system configurations – along with varying levels of support for variants – will result in different user experiences.”

COMMENTS:

- This depends on the provided solution and the techniques used to overcome these problems.
- In fact, these outlined problems are present for long period of time even for ASCII. For example, user names, file names, passwords, etc can be case sensitive or not depending on the OS or application. Users know these problems and live with them. It has not been solved to this time!

Pg 15, Pa 3 & 5:

- *Registrant applies for a domain name in Arabic language, which is then considered Primary*
- *Variants are created due to digits which are semantically same as ASCII digits and letters which are culturally confused due to local conventions*
- *Variant due to digits are automatically activated*
- *Variants due to writing conventions are blocked by default, but may be activated by the request of the registrant*
- *Blocked variants are not available for other registrants*
- *Variants can be activated at any time through the registrar*

- A registrant may activate up to **five variants** at a time
- All variants are associated as a set to the primary and may not be separated

...

Similar policies are adopted for other Arabic-language IDN ccTLDs, with some variation. For example, Qatar IDN ccTLD considers the characters *ﻩ* and *ﻩ* as variants in addition to other characters considered variants by dot-Emarat. **Jordan IDN ccTLD allows up to three SLD variants to be activated**"

COMMENTS:

- These differences are normal and it would not cause any confusion provided that the other variants that are not activated are blocked from registration by others.
- The number of allowed variant activation is determined by the business needs of the registry. It is solely their decision based on their business needs.
- Examples:
 - The number of accepted name servers per domain name (some accept up to 5 and some 10 and some are open).
 - The number of contacts for each domain name, some registries support only one admin contact, one technical contact, one billing contact, while other registries support multiple admin contacts and do not support billing contacts.

Pg 15, Pa 6:

*"The IDN ccTLD for Saudi Arabia offers position-level variants for each character [22]. Though the technical solution is more complex, this enables them to **offer many more labels to their potential registrants**. The IDN policy [25] regulates the labels. The policy also states that it "may establish rules and procedures to resolve the problem of character variants with other Arabic script based languages (e.g., Persian, Urdu)." As explained in supporting"*

COMMENTS:

- The solution used by SaudiNIC is based on the "Master Key Algorithm" that was developed by SaudiNIC and it provides a comprehensive and systematic solution. It has many advantages and addresses many problems that were outlined in this report. It is not really a complex if you know that it provide a very workable, adaptable, and configurable solution to a registry that would like to support many languages communities in gradual fashion without mixing variants regardless of the supported languages.
- One of the key benefits of SaudiNIC solution it minimizes the number of variants as it only generates the actual variants which confusingly similar to the needed domain name based on position-level variants for each character. Therefore, the number of generated variants is always the minimum.

Pg 16, Pa 2:

“These TLDs are already offering variant registrations at SLD or below in the IDN ccTLD, except Saudi-NIC, which is currently only offering a single (primary) label at SLD and lower levels, and intends to offer variant resolution if and when the variants are enabled at TLD3”

COMMENTS:

- Maybe I should be blamed for this incomplete information as I provided them without explanations.
- SaudiNIC has 2 sets of variants
 - At the language level (similar to other Arabic countries)
 - At the script level
- Currently we provide variants at the language level (up to 10 variants can be activated) and this was provided since we provided IDN registration.
- The script-level variants is ready but not rolled out to the registrant until we get the IDN TLD variant from ICANN.

Pg 19, Pa 3:

“• As this IDL set may be large, the registries set limits on how many variants may be activated. Chinese registries allow for all-simplified, all-traditional and one additional user-defined variant to be activated, limiting the total to three variants. For Arabic registries the limit on the number of active variants varies between three to six labels, arbitrarily chosen by the registrants. There is also a proposed three label limit for Devanagari IDN ccTLD. Interestingly, none of the IDN ccTLDs for the scripts restrict the activation to a single label, for promoting a better user experience. For the Canadian French variants, there is no limit on the number of variants.”

COMMENTS:

- As I said before we support up to 10 variants besides the primary name. The registrant can later ask for more if needed and it would be granted if it is a valid request.

Pg 19, Pa 3:

“Though the Chinese-script registries share the same table and have cohesively defined variants across all languages using the script, the Arabic script community exhibits many differences within the Arabic language and across the languages using the Arabic script. For example, even though all Arabic-language registries use the same reference language table (RFC 5564), they implement it with slight variations. For example, Jordan has extra variants defined for the letter ﺝ. Furthermore, IRNIC and Saudi-NIC intend to implement positional variants (using ZWNJ), whereas others are implementing character-level variants (by not allowing ZWNJ). Registries serving other languages using Arabic script, e.g. Iran IDN ccTLD, use a different language table. Registry for Devanagari defines a single language variant table for multiple languages supported by the TLD, which is the superset of all the code points in these different languages. Thus, there are three different strategies, with

Chinese script strategy (for a single table for all languages supported by Chinese script) being most consistent from the end user perspective.”

COMMENTS:

- In the contrary, SaudiNIC has a very strong position against using ZWNJ in domain names.
- Our position is well known to some of the authors! And documented in the Arabic Script Case Study report (<http://www.icann.org/en/topics/new-gtlds/arabic-vip-issues-report-07oct11-en.pdf>)
- Allowing ZWNJ in domain names is a very serious problem and threat to our users.
- ZWNJ causes some security, stability, usability, and reachability problems, and hence mistrust of IDNs.
- Here are some of the arguments that the Arabic Script Case Study report outlined against allowing ZWNJ (page 7 of <http://www.icann.org/en/topics/new-gtlds/arabic-vip-issues-report-07oct11-en.pdf>)
 - At Script level, the ZWNJ is considered by UNICODE to be an invisible join control character and listed in the "Unicode Security Considerations" document, which warns that incorrect usage can expose programs or systems to possible security attacks. This is especially relevant for IDNs.
 - The ZWNJ in some cases is not visible to all users (e.g., U+0637, U+0638, U+069F, U+06BE, and U+06FF). A comprehensive analysis of Unicode Arabic Script Code Charts is needed to find any additional cases. This process should be repeated as the Unicode gets updated.
 - The ZWNJ concept and behavior are not known to many Arabic script users, who do not use it or know how to type it.
 - ZWNJ is not conveniently available on the keyboard, where typing it requires multiple simultaneous key-presses, which is complicated for users. ZWNJ is also inconsistently placed on keyboards across various operating systems. In addition, it is not available on many keyboards, making it difficult for people to use ZWNJ when, for example, they travel.
 - The users may not be able to type a domain name as they may think it is a <space> not ZWNJ, which may lead to reachability and usability problems, and therefore, mistrust of IDNs.
 - Based on the ZWNJ Contextual Rule (RFC 5892 Appendix A.1) for handling CONTEXTJ labels under the current IDNA2008, the Rule implementation does not totally resolve the non-visibility problem particularly in some cases as discussed above (e.g., U+0637, U+0638,U+069F, U+06BE, and U+06FF).
 - It is not one with the general category of {Ll, Lo, Lm, Mn}, as per the requirement defined by the gTLD Applicant Guidebook (v 2011-09-19,Module 2, page 2-13, Section 2.2.1.3.2, Part II, Item 2.1.3.).
 - Root policy should be more conservative than labels for other levels.
 - Use of ZWNJ may cause additional bidirectional display issues.
 - Hyphen can be used instead of ZWNJ to break a string into ligatures

Pg 23, Pa 6:

“For example, users will expect that if two strings are variants at TLD level, they are also variants at SLD level, and if these strings are activated at TLD level, they can also be activated at SLD level (and not blocked). Further,

if two labels are variants in a script and are allowed for activation in one registry supporting the script, user will also expect them to be variants and allowed for activation by another registry supporting the script.”

COMMENTS:

- You cannot guarantee similar behavior across TLDs even under the ASCII TLDs.
- It depends on policy, practice, prices, etc.
- Each language has its own characteristics and needs.
- Existing practices (within ASCII TLDs) are also different from registry to registry. For example, company domain name can be found under co.uk in United Kingdom and they are under com.sa in Saudi Arabia.

Pg 27, Title1:

“5.1. Challenges with the Use of Variants”

COMMENTS:

- This section frightened and horrified the reader from variants.
- It did not illustrate the problems and security issues if variants are not supported or activated.
- The need for variants is quite important and crucial.
- This Section listed several challenges related to active variant TLDs. Even though they appear accurate and true from the first look, but they are not caused by variants. Some of the challenges already exist in the current (IDN or ASCII) domain names.

Pg 27, Pa 1:

“The challenges presented in this section are primarily concerned with the end user. Some end users may be able to handle variants more proficiently. However, the current challenges are listed in the context of novice users, though familiar with the script. These challenges are equally relevant for application developers who try to make the interface friendly for end users. Variants of IDNs are a new concept and will surely challenge application developers to find mechanisms to make them easily and securely available to end users. Many of the challenges are generic enough to be applicable across many other users who will be interacting with variants, including registrants, sales and marketing staff with registrars and registries and even more experienced users. For example, the first challenge, related to finding the complete set of variants, universally impacts all users.”

COMMENTS:

- Variants are used and supported from day one of implanting DNS. Upper case ASCII domain names are variants to the corresponding lower case domain names. Luck it was solved at the protocol level (not application level as IDN) and hence was not noticed by the users.
- Even if the IDN variants concept is new it should be encouraged and fully supported and not frighten and horrify the reader.

Pg 27, Pa 3:

“There is no single place from which different kinds of users can determine the complete set of variants for a domain name, including TLD”

COMMENTS:

- IANA has repository for language tables used by registries.

Pg 27, Pa 6:

“Though end users generally interpret the use of labels within the context of a language, current discussions suggest that variants for TLDs will be defined in terms of scripts. This may impact user expectations.”

COMMENTS:

- In many occasions we recommended determining the accepted characters (code points) for writing an IDN label (including variants) should be based on languages rather than script table. Many of the confusions and risks outlined in this report can be avoided through adopting language table for each registry. Users write and read a language and not a script.

Pg 27 & 28, Pa 8:

“For example, in Arabic script U+06A9 and U+06AA are considered distinct in the Sindhi language, but may be considered as variants as these are stylistically different characters in other languages (e.g. Urdu and Farsi) (Arabic Case Study Team, 2011). This would mean that for Sindhi speakers, two distinct labels may be considered variants, which is not intuitive for users.”

COMMENTS:

- This example illustrate the need to indorse our recommendation to determine the accepted characters (code points) for writing an IDN label (including variants) should be based on languages rather than script table.

Pg 28, Pa 1:

“User considers two labels as variants of each other due to a certain linguistic context, but the labels are delegated as independent IDN TLDs.”

COMMENTS:

- Users do not care how variants get implemented but rather they should work and work correctly.

Pg 28, Pa 2:

“Users expect the independently delegated TLDs to be unique. However, there is a possibility that two independently delegated TLDs are considered variants of each other by users of a linguistic community. This situation may arise if a linguistic community is not considered in the formation of a LGR for their script or the current process independently delegates two IDN labels as ccTLDs and/or gTLDs which are later determined to be variants by the LGR development process.”

COMMENTS:

- Not clear at all! And it looks odd to happen.

Pg 28, Pa 5:

“For example, for a Sindhi-language based SLD, the use of U+06A9 and U+06AA may result in two different labels, but as they are considered variants at the TLD, end users may also consider them as variants at the second-level.”

COMMENTS:

- This is solve by using language table rather that script tables
- Our solution “Master key algorithm” handle this problem

Pg 28, Pa 7:

“This inconsistency of support may be caused by many different factors, such as different software configurations (including legacy systems). This includes different operating systems, locale settings, fonts, rendering engines, and applications, to view the domain names. Further, some of the recently encoded Unicode code points may not be supported by technology. Limitations in processing and storage capacity (e.g. on mobile platforms) may further aggravate the challenges.”

COMMENTS:

- This is a general comment that is not particular to variants. Content provider also affected.

Pg 29, Title1:

“5.1.6. Variants cannot be input by the user”

COMMENTS:

- It is not necessary for the end user to know or to be able to type all the variants. He/she needs to be able to write one form of the variants (using his/her keyboard) and reach the ultimate resources (e.g., web site). It is the responsibility of the registry to make this happen transparent to the user.

Pg 29, Pa 3:

“For example, a user who has an input method set for Simplified Chinese may not be able to input Traditional Chinese label, and vice versa. So a user may not be able to access relevant content. This may be a more significant issue while using domain names in mobile phones.”

COMMENTS:

- This is a general comment that is not particular to variants. Input to a search engine .

Pg 29, Pa 4:

“There may be instances where users need to identify a specific variant, such as to log into a system. However, in many cases, the variants may be visually very similar or the same (when displayed as U-label) and A-labels may be mnemonically intractable. Thus, an end user will not be able to determine the specific variant needed in a specific case and will have problems in using such systems.”

COMMENTS:

- This is a hypotheses assumption.

Pg 29, Pa 5:

“A user may need to identify a specific (e.g. primary) variant of a variant set. In many cases this may not be feasible because variants may be visually similar or exactly the same when displayed as U-labels. As the A-labels are not good mnemonics, the user will not be able to distinguish the variants from each other.”

COMMENTS:

- This is a hypotheses assumption.
- Why you need to distinguish undistinguishable?
- The registrant registers the domain that he/she can type. Other variants are either blocked or activated.

Pg 29, Pa 6:

“For example, a user may input an email address for signing in to an e-commerce website. The user would need to remember the specific variant to login.”

COMMENTS:

- This is a memory problem not variants 😊
- If you have multiple email (yahoo, gmail, hotmail, company, ...) which one you used for registration?
- By now I feel that the author try to find anything to discourage the use and support of variants.

Pg 29, Pa 7:

“Many websites ask users to identify themselves through their email addresses or another identifier that includes the domain name. If variants are introduced, users may be able to input different variants of the domain name (knowingly and, in most cases, unknowingly as these variants maybe visually identical). Thus, users may experience systems that do not work as expected.”

COMMENTS:

- [See previous comments](#)

Pg 30, Pa 1:

“If a user deletes certain domains from history, the variants of the domain name accessed by the user (which may have been accessed without explicit user knowledge, e.g. by clicking on a visually same variant) may not be deleted. This will negatively impact the privacy of the user.”

COMMENTS:

- [Similar to multiple domain names that point to the same website.](#)

Pg 30, Pa 2:

“Search techniques may not consider variant domain labels as related, and thus do not find relevant web pages against a user query.”

COMMENTS:

- [Similar to multiple domain names that point to the same website.](#)
- [This will get better by increasing the usage of IDNs](#)

Pg 30, Title2:

“5.1.11. Search rankings unpredictable”

COMMENTS:

- [If an entity has multiple domain names due to protection against fishing or domain squatting would face some of the same challenges](#)
- [Search engine optimization is complex and dynamic and is not related to only variants.](#)

Pg 30, Title3 & Pa 9:

“5.1.12. Search optimization affected by variants

.....

As most web frameworks do not support variants or multiple domains, software developers may use the HTTP “Move permanently” response to redirect multiple domains to the same web page. This may significantly increase the number of redirects, lowering the search engine ranking.”

COMMENTS:

- This is a improper practice.

Pg 31, Pa 3:

“For example, users may expect that if `www.SLD.TLDvariant1` and `www.SLD.TLDvariant2` work, then email addresses `TLDvariant1@SLD.TLDvariant1` and `TLDvariant2@SLD.TLDvariant2` should also work. However, `TLDvariant1` and `TLDvariant2` before the ‘@’ sign in the email may not be equivalent. Similarly, the following is not considered equivalent in the URL `http://www.SLD.TLDvariant1/prg?t=TLDvariant2`. This will confuse the user who is not able to distinguish the two uses of the labels.”

COMMENTS:

- It will work if configured correctly.
- Misconfiguration is not part of variants!

Pg 31, Pa 4:

“Web sessions may need to be re-established if the same website is accessed through a different variant. This may confuse end users, especially when the variants are the same or similar looking.”

COMMENTS:

- This is a hypotheses example.
- User (if needed) will type the website address once.

Pg 32, Pa 7:

“For example, it may not be possible to have the same SLD variants to be activated in the same way across the various TLDs of the same script due to differing variant handling policies (activation status over time, primary label definition over time, pricing, etc.). Registrants may find the process for registration, renewal, de-activation and deletion of variant sets so confusing that they may be discouraged from using IDNs.”

COMMENTS:

- This is done now with ASCII TLDs.
- Each registry has its own way to manage its registry, determine the accepted label length, prices, reserved list, dispute policy, etc...

Pg 32, Title3:

*“5.2.3. **Inconsistent association of ASCII and IDN TLDs**”*

COMMENTS:

- This is done now with ASCII TLDs.
- Each ccTLD has different 2TLDs (com, co, edu, ac, ...)

Pg 33, Pa 3:

*“**Technology to support the registration currently does not handle variants**, which may make variant registration challenging and difficult.”*

COMMENTS:

- Welcome to the IDN world.

Pg 33, Pa 7:

*“**New registrants and users of IDNs and variants may be monolingual and may only understand a single non-Latin script. Registrars that are providing interfaces may not be able to predict the language of the registrant for a given TLD and thus may not be able to localize the interface. This may make it difficult for registrants to register IDNs and its variants, particularly given the inherent complexity of variants.**”*

COMMENTS:

- This is driven by business needs. It will come as the IDN business grows.

Pg 33, Pa 8:

*“For example, a potential Arabic script IDN TLD registrant may speak Urdu, Pashto, Arabic, Farsi, Sindhi, or any other of the many languages using Arabic script. A gTLD may cater to many of these communities. In some cases, even a ccTLD may cater to multiple languages, e.g. the Arabic script IDN ccTLD for Pakistan and India. **A registry or registrar may not know which language to offer within the registration interface, or even offer support such a language.**”*

COMMENTS:

- This is driven by business needs.
- This is a place for competition between registrars to serve certain communities.

Pg 34, Pa 1:

“Variants are not part of the registration data and services infrastructure at this time. Therefore, registrants will not be able to define and access data related to variants consistently across registries, creating usability and security challenges.”

COMMENTS:

- This is not accurate
- For example, SaudiNIC display the originally registered domain name when you search for one of the variants regardless if it is activated or blocked.

Pg 34, Pa 4:

“Due to potentially large set of variants and their confusingly similar strings, protecting trademarks may become much more complex and costly.”

COMMENTS:

- Not really. By registering the primary domain name, variants are automatically blocked from registration. They can be activated only by the registrant.

Pg 34, Pa 5:

“A domain which may be seemingly unrelated may have a variant which can be exactly the same or similar to an existing but different trademark. This may be difficult to anticipate and track, especially due to variety of implementations across TLDs, due to complexity of variant management processes for both SLDs and TLDs, and due to lack of support of registration data and services for variants. With many more TLDs and potentially many more variants and strings similar to these variants, tracking and protecting trademarks will be an uphill task.”

COMMENTS:

- The new gTLD has the same problem.
- Policy and implementing variants will provide the right protection (see previous comments)

Pg 35, Pa 5:

“For example, a webmaster configures the web platform for all activated variants (tens of variants are possible). But because each variant may require a separate, unrelated configuration, the process is both tedious and error-prone.”

COMMENTS:

- Same for multiple domain names with different spelling

Pg 35, Pa 6:

“System configuration software may display domain names only in ASCII format. This means that IDN labels will be displayed at A-labels (not as U-labels), making it challenging for system configuration personnel to manage them.”

COMMENTS:

- This IDN problem not variants

Pg 35, Pa 7:

“Many of the operating and configuration systems may require domain labels to be written in ASCII (A-label), as they may not fully support non-ASCII file formats and filenames. This means that even if variants can be supported, they may still need to be configured as A-labels. However, A-labels, due to the nature of the algorithm that produces them, generate intractable ASCII strings that cannot serve as mnemonics.”

COMMENTS:

- We cannot blame variants. IDNA is the one to blame.

Pg 36, Pa 1:

“mgbai9azgqp6j” for the latter string. So it is difficult to use either representation conveniently to configure systems.”

COMMENTS:

- Why?

Pg 40, Pa 3:

While activating variants may better serve specific language communities, this report does not make any specific recommendations on how variants should be technically implemented in the DNS or even whether they should be implemented at all. Understanding that each additional active variant may add complexity and/or security risks for the registrants and end users, the recommendations presented here generally seek to block/withhold a maximum number of variants while activating a minimal number of variants.

COMMENTS:

- Why?
- The recommendation should be based on community’s need and user protection not minimizing the number of activated variants.

Pg 41, Pa 1:

“A TLD label application may have many variant labels. The applicant may desire one or more of these variant labels for activation. The minimality principle discussed earlier states that given the inherent complexity of supporting and using variant labels, a conservative approach to adding variants is advisable. Minimal necessary active variants will provide a better user experience, as they are easier to configure, manage, monitor and navigate.”

COMMENTS:

- Minimizing is not a goal if there is a need. Variants are determined by the local communities regardless of their sizes or numbers. They are bound by the need and not limited by the size. Variants solve and address some security issues and therefore are needed regardless of their numbers.

Pg 41, Pa 6:

“All requirements for a TLD application approval process also apply to the approval of a variant TLD. These include, for example, requirements for GAC and public comments on the label, string similarity evaluation, DNS stability evaluation of the variant TLD label, etc. ICANN must document this process.”

COMMENTS:

- Unacceptable process
- Why redo everything? The initial string is already approved and assigned so we need a light process just to activate the other strings (variants). Why GAC? public? ...

Pg 41, Pa 11:

*“a. Root zone LGR
b. State of each variant (activated, withheld, blocked, etc.)of each allocated TLD
c. Second-level LGR submitted for each TLD”*

COMMENTS:

- Who define variants at the root zone LGR? And how?

Pg 42, Pa 5:

“Such rules should avoid integrating complex script or language dependent deviations to minimize inconsistencies. Any deviation necessary for a particular script or community should be clearly justified and documented in the root LGR.”

COMMENTS:

- One size fit all DOES NOT work for languages.

- You deal with natural languages that each one has its own characteristics. In Chinese language, for example, you can have a meaningful word just by typing one Chinese character. For example, the limitation for 3 characters in TLDs that may be suitable for ASCII is not suitable for Chinese language.

Pg 42, Pa 8:

“The code points allowed for LGR must include only those minimally needed by a particular script community. For example, the repertoire should not include dead scripts and code points representing archaic characters that are not currently in use by a script (as per IDN P2.1).”

2. If the community cannot agree on the need of a code point, the default decision must be not to include it in the repertoire.”

COMMENTS:

- Needs come from language communities not script.
- There should be other way to make the decision rather than just “not to include”

Pg 42, Pa 11:

“Code points in the LGR for the root should be based on script. Language-specific code points should be minimized, and any such code points should be explicitly justified. This may result in increasing the variant labels but will promote consistency of use across global end users.”

COMMENTS:

- Why not language? No one speak or write a script.
- Unjustifiable recommendation. The authors already pointed out the problem of defining variants based on script (see our comments above on Pg 27, Pa 6).

Pg 43, Pa 1:

“Variant rules not motivated by security reasons must only be allowed if there is significant community need. These should be accepted based on documented community need.”

COMMENTS:

- Not accepted ... very complex process and intensive and might not reach conclusion!

Pg 43, Pa 2:

*“As TLDs – particularly gTLDs –are used by a global end user community, variants for TLDs must be defined for a script. **Language-based variant rules should not be allowed.**”*

COMMENTS:

- Why?

Pg 43, Pa 3:

*“A simpler variant rule is preferable to a more complex variant rule, in instances in which there are options available; **for example, 1:1 code point variant rule is simpler than 1:many code point variant rule; context-free variant rules are simpler than context-sensitive variant rules, etc.**”*

COMMENTS:

- Provided that it will not affect or alter the registry’s business.

Pg 44, Pa 5:

*“The applicant must have a clear and consistent linguistic policy for the second-level (including LGR and variant state life cycle for second-level) **that is not in conflict with the root-level LGR and state life cycle.**”*

COMMENTS:

- This is not valid request as the root usually conservative while 2nd level are more option available.
- Example, numbers might not be allowed at the root-level but it will be available at the 2nd level.
- Out scope of the report.

Pg 44, Pa 8:

“a. A registration system supporting the registration and activation of variants
b. A DNS zone generation supporting variants
c. A registration data query system using **the standard protocol supporting variants**”

COMMENTS:

- Which protocol?

Pg 44 & 45, Title2 & Pa 9 & 1:

*“6.1.6. ICANN should require IDN TLD registries with variants to apply the relevant (script) subset of the root zone LGR and state life cycle for variants across second-level domain labels. **Deviations should be justified**”*

Minimal, simple, and consistent solutions across the DNS tree enable software developers to more consistently and easily integrate support for variants across applications that interact with the DNS. End users, in turn, benefit from a consistent level of support for variants across applications. **Conversely, any deviations by registries risk generating variant sets that are not compatible with general software applications.** Therefore, registries must be required to adopt the relevant portion of the root-level LGR at other levels in the DNS, and carefully consider and justify any deviations to help ensure a more consistent and predictable experience for registrars, registrants and, ultimately, end users. Any additional code points not in the root LGR (e.g., digits, hyphen, etc.) and corresponding rules should be clearly documented **Registries should also be encouraged to adopt at the second-level the same state life cycle defined for variants at the root** so that users have a consistent experience across the DNS tree. Any deviations should be carefully considered and justified”

COMMENTS:

- This is not valid request as the root usually conservative while 2nd level has more option available.
- Out scope of the report. The report must limit its scope solely to IDN TLDs.

Pg 46, Title2:

“6.1.9. ICANN must develop consistent registration data requirements for variants at root **and other levels**”

COMMENTS:

- Out scope of the report. The report must limit its scope solely to IDN TLDs.

Pg 46, Title2:

6.2. **Recommendations to a Registry** that Offers IDNs for Scripts that have Variants

COMMENTS:

- Most of the recommendations in this section, even though they are still recommendations but they are out scope of the report. The report must limit its scope solely to IDN TLDs.

Pg 47, Pa 5:

“The registration of a **variant label must not be automatic**, but initiated on the request of a registrant, explicitly specifying (i) the variant label and (ii) the state for which the variant is being requested (delegated, allocated but not delegated, etc.). By default, all variants should be withheld (and un-allocated).”

COMMENTS:

- No need to make the request complicated. Automation does not mean assignments without policy. The process can be automated and adhere to certain policy suitable to the root.

Pg 48, Title1:

“6.2.3. Registry that supports variants should apply the LGR developed for the root across lower-level domains. Deviations from the LGR should be publicly documented and justified”

COMMENTS:

- This is not valid request as the root usually conservative while 2nd level has more option available.
- Out scope of the report. The report must limit its scope solely to IDN TLDs.

Pg 48, Title2:

*“6.2.4. Registry that supports variants **must implement**, to the extent possible, state life cycle for the second-level variant recommended by ICANN”*

COMMENTS:

- Out scope of the report. The report must limit its scope solely to IDN TLDs.

Abdulaziz Al-Zoman