ICANN72 | Prep Week – Root Zone Label Generation Rules Update
Thursday, October 14, 2021 – 11:00 to 12:00 PDT

PITINAN KOOARMORNPATANA: Thank you. Good morning, good afternoon, good evening, everyone. Welcome to the root zone LGR update session during ICANN 72 prep week. Today, we have this agenda. I will just give a quick overview, five minutes on the background of the root zone LGR project, and then we will move forward to our key speakers today. We have Mats Dufberg from the Latin script generation panel members, Hiro Hotta from Japan for the Japanese generation panel, chair, and then we have Michel Suignard from the integration panel to talk about the next version of the root zone LGR.

So let me move forward with the first part, an overview of the root zone LGR. So a brief history of the project. Around 2010 also, the ICANN community identified the needs of the variant top-level domains and for that, the community comes together and works on the integrated issue report. At that time, it was to explore if they implemented variants into the root zone, what would be the issues. So it was the issue study, not really the solution back then.

In that report, there were six scripts, including Arabic, Chinese, Cyrillic, Devanagari, Greek and Latin, so cover multiple types of script used in the modern languages.

Then the report was developed and published in 2012, and in that, it's also identified that they should have the mechanism to define what is the variant for each script. And for this mechanism to identify should be

Note: The following is the output resulting from transcribing an audio file into a word/text document. Although the transcription is largely accurate, in some cases may be incomplete or inaccurate due to inaudible passages and grammatical corrections. It is posted as an aid to the original audio file, but should not be treated as an authoritative record.

a single source because eventually, all these TLD will go to the root zone.

The community identified root zone LGR as the mechanism to define and the LGR [positions] to develop the root zone has been approved by the Board in 2013. Since then, the community has formed generation panels and work on the root zone LGR for the respective scripts.

In parallel, ICANN Org also developed the recommendation of managing the IDN variant TLDs. This recommendation has been approved by the Board in 2019. It resolved that the GNSO and ccNSO should take this recommendation in the policy development and [inaudible] integrate the use of root zone LGR.

Then in 2020, the ICANN Board also resolved that GNSO and ccNSO take into account the recommendation of the technical utilization of the root zone LGR and then resolved in 2021 that GNSO also publish the report for the SubPro, which also incorporate the use of root zone LGR and for the next round.

So, how does the LGR work? LGR is like a formula of how to form the labels. So it’s the label generation rules. So when we put these rules integrated together into the root zone, it'll be readable by machines, so it'll be readable by the tool, and then for that, we can mechanically identify whether the label is valid or not, and if it is a valid label, it also can identify the variants of it which can be allocatable or blocked as well.

The mechanism by the LGR procedure, they work in two steps. First will be the work from the generation panel which developed the solution for the particular script and then once the generation panel for each script has finalized their solution, it'll be submitted to the integration panel. Integration panel will consider the proposal from the generation panel or the GP and then it could be multiple time of consultation between the GP and IP. And once the solution is successfully reviewed and accepted, it'll be integrated into the root zone, and that can be used to validate the TLD.

And this is the summary of the work by the GPs. So for these 28 scripts on the screen, we have 17 generation panels covering 26 scripts for now. as you can see, the work started from 2014 and it's multiple years of effort. Right now, many of the GPS already finalized their work, and currently, we have Japanese and Latin script which is ongoing public comment proceeding for now. Myanmar is the one still finalizing their proposal for the public comments to come.

With this, this is the timeline of the root zone LGR. So starting from 2013 for the call for panels, then we have the panels active and working from 2014. Over time, we have the script finalized, that work, and integrated into the root zone incrementally.

To date, we are at root zone LGR version 4 which we have 18 scripts in total, and we are anticipating that in 2022, we will have integrated eight more scripts and this will mean a total of 26. That's where we are today, and today we have members from Latin GP and Japanese GP to go over their solution, and also would like to invite everybody to participate in the public comment proceeding as well.

So let me hand it over to Mats Dufberg from the Latin GP. Over to you.

MATS DUFBERG: Thank you very much. Next slide, please. So this is the parts of my presentation, and I will talk about the proposal that we have out for public comment. So these are the main parts I will talk about. Next slide, please.

So we have the Latin script proposal out for public comment, and comments will be accepted until 23rd of November this year. There is a link to the proposal, and we encourage everyone to go in and look at our proposal, and we will of course consider all input of minor comments and major comments. All is welcome. Next slide, please.

This presentation is a walkthrough of the proposal, and we hope that this will lower the threshold for you to go and read it and provide comments to us. So focus is on the main document and its dependencies, and the LGR XML file is a normative document, I would not talk very much about that.

So the chapters I don't discuss here are just short chapters with general information that are not worth discussing here. Chapter two in the proposal defines the limitation of the script by this proposal. The first limitation is that the proposal cannot include any character not included in the so-called maximum starting repertoire. So the MSR is a subset of the IDNA protocol valid codepoints, which is a subset of

Unicode. So this MSR is therefore a subset of Unicode and that is what we are bound to keep the proposal within.

The MSR has been defined by the integration panel. So during the process, there were a few characters that we proposed as an addon to the MSR and IP, the integration panel, added those characters to a new revision of this MSR. MSR is a living subset of Unicode, it means that it can be increased, and I guess it can be decreased too.

We have, during the process, proposed a few characters that were added. Chapter 4 describes the work process of our work. We have to find out what characters to use for our proposal. We decided to use the so-called EGIDS scale, which is explained on the next slide. From level zero which is international languages to level four, educational. Then we also added level five, developing, with at least 1 million speakers. So those are the languages. Next slide, please.

Here we see the EGIDS scale, the expanded, graded intergenerational disruption scale which we used. Almost all languages in the world have been graded in this scale. So we used that to select the languages to look at. So zero to four and then additionally, level five where we used languages of a population of users of 1 million speakers. There are other levels too, but excluded here.

Appendix 6, there's a complete list of all languages selected. I don't have the number here, but it's quite a lot of languages. So for each language included, we have the language name. It's 212 languages that we have in our selection. In that list, there is a language name, the ISO language code, and also the EGIDS level for the language.

So for each language in the list, we identified all characters used for writing, and so we saw what characters they use to cover these languages. So the idea is that these 212 languages should be able to create top-level domains for the words in the language. The character set of each language is not documented [in the] report, but there are references to each language found in chapter nine. Candidates for variants were also identified. More on that below. Next slide, please.

So the repertoire of Latin script proposal is naturally bas ed on Unicode codepoints, because that is the limitation. In the simplest case, a codepoint is just a character such as letter A. A codepoint can also be a modifier mark. Then it's used in combination with another code point to form a character. On the slide, there is an example of the J with tilde on top which is formed by a combination of two codepoints.

In many cases, there are precomposed codepoints with the base character and an accent such as A with acute that is shown on the slide. So that is only one codepoint. So the principles for including or not including a character identified in the language are spelled out in the introduction in chapter five. Next slide, please.

The proposal for Latin lists 218 characters. 197 characters are of just one codepoint, so that's the simple kind. 21 characters are formed by a sequence of two or more codepoints, so like a base character plus a mark, or two marks. So for each character, there is the codepoint that a character consists of, either a single codepoint or a sequence.

The language or languages that use that character writing, not necessarily all languages but enough languages to attest that this character is used for languages selected. And references to the alphabets of those languages are also in the list. For characters a to Z, we don't have any languages attesting those because those are included by default. Next slide, please.

So the repertoire is also one of the main parts of the XML file. The repertoire in chapter five is sorted numerically by codepoint, but if you want to look at the same characters but grouped by glyph shape, so like all A's with different accents on, that is found in Appendix C. So it's the same list but sorted differently. Next slide, please.

In section four of chapter five, we list excluded characters. Those are characters that were attested in at least one selected language but cannot be included because they do not belong to MSR. So the MSR is this preprocess, so makes the upper limit of what can be included. So the Latin GP cannot include any character not in MSR.

So if you find a letter that you think should be included but that is not part of MSR, this is nothing that Latin GP can change. It must go to the update of MSR first. Next slide, please.

So I talked about the repertoire. The next part are the variant sets, which is an important part of the proposal. Chapter six covers that. So, what is a variant? Well, a variant set consists of two or more characters that are in some way perceived as being the same, treated the same in some way. They could be the same shape or it could be that it's used interchangeably for some part or the whole part of the script community.

So when you have a variant set, then you cannot use both characters independently. And in most cases, if you use one of the characters, the other is blocked in the same context. So for Latin script, most variant rules lead to locked variants.

In-script variant sets have members from the same script, the Latin script, but we also have cross-script variant sets with members from different scripts, mostly Latin, Cyrillic and Greek, and some sets are a combination. Next slide, please.

So chapter six, together with appendices, D1 to D9, contains the principles for variant sets and data analysis of variant sets and candidate variant sets. So you can study that to see why we have come to the conclusion that these are variants.

So two variant sets are special. They're related to older IDNA version 2003, and they also include permitting allocating the other variants. And the two sets relate to ... one is sharp S and SS, and the other is dotted I and dotless I. So those two sets are special in the proposal. Next slide, please.

And then finally, in appendix E, we have a list of visually confusable—sets of visually confusable characters. So those are not variants, because we don’t suggest them to be variants. They're not part of the formal LGR XML file, and they're there for reference for anybody that is doing analysis of visual similarity between script. Thank you. Yeah, this is the process, so we are out in public comment right now and we hope to submit the final version in January next year. So please read and provide comments on our proposal. I don't know how you deal with questions.

PITINAN KOOARMORNPATANA: Any question in the chat, or if you'd like to speak, you can also raise your hand as well. Okay, seeing none, let's move forward first and then if you have any question for Mats or for Latin GP, we can come back as well. Next, I'd like to invite Hotta-san to give the overview of the Japanese root zone LGR. Hotta-san, over to you, please.

HIRO HOTTA: Thank you, Pitinan. Hello all. My connection is not very stable. Sorry if I lose connection. But in a couple of minutes, I can resume, I hope. Okay. Next slide, please.

This is the members of the Japanese script generation panel. Much the same situation as with other GPs, we have members from various [views.] Next slide, please.

This is the overview of the script and language. I think it's very unique that the Japanese language has three scripts, even with ASCII and numbers, but those three scripts are kanji, hiragana, and katakana. Those characters can be mingled in any order in a word. And as the LGR characters defined in JIS, Japanese Industrial Standards level one and two, we selected [them] as the repertoire of the Japanese LGR.

So as kanji has many characters, we have around 6000 characters in our repertoire. Of course, it's in the MSR. So we have three kinds of scripts, and one of them, kanji, is already used in the Chinese and Korean languages. This makes some difficulties in defining the variants. I will talk about it later.

Other than kanji, we have hiragana and katakana. Hiragana is mainly used as suffixes to kanji to complete a full reading of the word, for adverbs, for conjunctions and to rewrite difficult kanji into forms of easy writing and reading. For example, for five-year-old children or so.

Katakana is mainly used for other kind of words. Katakana is mainly used to represent long words from foreign languages and onomatopoeic words.

Basically, all Japanese characters are regarded as independent, but in Chinese and Korean languages—as I said, Chinese, Korean, Japanese language share some kanji character—some sets of kanji characters are regarded as variants when two or more characters have the same name and pronunciation.

And some people in the Japanese language community believe that some Chinese or Korean variants are regarded as variants in the Japanese language as well. Next slide, please.

So this is the overview of the root zone LGR for Japanese. Repertoire, as I said, 6000 characters, and variants—no intrinsic variants for Japanese, meaning stemming from the same name and pronunciation. But from visual identicalness, 12 sets of variants are defined, and this is a unique point, we accommodate kanji variants defined in Chinese and Korean

LGR. I think there are hundreds of them we imported from Chinese and Korean LGR.

This is the second important point: no variant labels other than the original label can be allocated. This is the hardest spot to decide in Japan, but we finally decided no variant labels.

And WLE, one very simple Japanese-specific rule. Any small kana, iteration mark or prolonged mark must not start a label. This rule is the same for ordinary Japanese words. Next slide, please.

Special attention point versus visual identicalness. So visual identicalness in Japanese script, we defined several of them. First is like hyphen and [prolonged minus], two marks, and Unicode consortium lists confusable characters between different scripts. And based on the list, we picked up the following ten pairs as candidates to be visually identical. So they are ten pairs. Words containing these characters are shown on the bottom right. Next slide, please.

And confirmation of visual identicalness of Unicode-based 10 pairs. So we did a field research to see if all of them are visually identical or not. So we examined with 40 testers. Among them, 20 read Japanese well and 20 don’t. So we scored each character pair whether they're identical or not, and the result was all pairs are rated less than 3.2. It means they can be confused. So we decided they should be variants.

The second field research was to see if there are additional visually identical pairs other than those ten pairs above. We did the research for around 176 diverse recipients and among them, 73 responded and the result was no pairs confused more than 3% of the respondents, so we decided that just the ten pairs in the Unicode confusable list should be the variants. Next slide, please.

So this is a second special topic. Basically, any combination of characters is allowed in Japanese labels, as I said. So this characteristic may make the number of variant strings large. And considering the definitions of many variants are imported from the Chinese and Korean LGRs. For example, this is a four-character kanji string, Keio Daigaku which means Keio University. It has three variant strings.

Actually, Keio University registers and uses all four variant second-level domains under .jp, so it means that the Keio University wants to use them all as a variant. But if Keio University is allowed to use them all, four TLDs are allowed to be in the root zone simultaneously. Four is not big, but such a rule may explode the size of the root zone when longer labels are considered. So reduction of the number of allocatable variant labels required to prevent explosion of the root zone size. Next slide, please.

So Japanese GP tried to devise various methods to reduce the number of allocatable labels, but any method can reduce the allocatable labels, less than five or something. So we finally decided that we allow valid applied for label only and make all variant labels blocked. So just one label is allocatable, which is an applied for label, and any variant labels from that are prohibited, blocked. Next slide, please.

So current steps, we are in the public comment proceedings, and it will be closed 16th of November. Thank you. Any comments, please.

PITINAN KOOARMORNPATANA: Thank you, Hotta-san. Any question in the chat?

SARMAD HUSSAIN: We have a question, but it is not directed to Japanese language. It is more general, so we will hold that and maybe after Michel has gone through his presentation, we'll then ask the question to the integration panel.

PITINAN KOOARMORNPATANA: Okay. Thank you, Sarmad. Thank you, Hotta-san. We will move to integration panel, Michel Suignard to give a presentation on the root zone LGR five. Over to you, Michel.

MICHEL SUIGNARD: Okay. Next slide, please. So the presentation will cover four topics as you can see here. First one is about the integration, concern of how we did the Chinese, Japanese and Korean LGR integration we're processing today.

[inaudible] slide concerning how to do processing a label [inaudible] and some interesting points on Armenian, Cyril, Greek and Latin script, and then the scope of root zone LGR five. So let me go into more detail on each of those. Next slide, please.

So this in fact is related a bit to what Hotta-san was just saying. One is a major task we are now facing, is integrating Chinese, Japanese and

Korean scripts through their own LGRs. So we have chinse, Japanese and Korean script are now complete. We have rulesets defined for the LGRs. The only one in fact open for public comment is the Japanese LGR. The other ones already went through the public comment a while ago. That was a very complicated task. Those LGRs are very big and they contain thousands of codepoints. We had a fairly complicated set of issues concerning variants on ... we come to that now or maybe during the questions later.

So since we got those three sets, we have been working on integration of those sets. Obviously, [we did not start the] completion of those, but we've been working with the various GPs to see how we could do it.

First, it’s interesting to note that variants in those LGRs are either created on their own into the LGR or inherited through integration. There's two sets of inheritance. There's one where you have at least two codepoints from the resulting LGR that are affected by integration from another repertoire or that has been affecting a lot Korean and Japanese, because in fact, the Chinese LGR is by far the one that has the most variants. That’s, as maybe a few will know, is due to the traditional and simplified duo of the Chinese repertoire. Many [inaudible] characters can be represented by a simplified shape of the character or the traditional one. So there are historical reasons.

Some of those simplified characters also do exist in some of the Japanese or Korean, the same shape, on which you also have a lot of variants between traditionals. It’s quite common that you have many traditional Chinese characters that are seen as the same, to use a definition used earlier.

So by far, the Chinese LGR has the most variants, because many of those variant sets have at least two members in Japanese or Korean LGR. Those have to be defined as well on those LGR. In fact, that meant that we had to basically ask the Korean LGR and the Japanese LGR to include those variant sets that add at least two codepoints from those who existed as part of a variant set for the Chinese LGR.

So that was the first part, basically making sure that all the variant sets, that at least two members inherited from the Chinese variant set were also defined in the Japanese and Korean LGRs. In those two, they're defined as blocked, where in the Chinese LGR, they may be sometime allocatable. That's another point of this system.

In addition to that, the Chinese LGR does in fact inherit some unlisted variants from the original submitted Chinese LGR. Those are mostly due to kana and hangul confusion because in the Japanese LGR, as you saw before, they created some variant sets between some kanji and kana characters. Those obviously need as well to be imported into Chinese because obviously, the Chinese character in many cases includes those characters as well, the kanji equivalent, if you will. Same for Korean. Korean is a bit different. Now it concerns hangul and hanja character—hanja is basically the Chinese character in Korean. So there's some confusion between hangul and some Chinese characters which also have to be inherited in the Chinese LGR.

For Korean and Japanese, a bit simpler. We only imported, like I said, one or two codepoints, but obviously, at the end, the Korean and Japanese have to include variant set that include one of the codepoint that is associated with other Chinese characters that are not included in their own sets. So you have these what we call other repertoire variants that do show in the Korean and Japanese LGR.

Result expected, obviously, when label collision would be determined via the merged LGR file. In every root zone LGR, we define what we call element LGRs which are in fact the integration of the proposed LGRs by GPs, and we also have another file which is called the merged LGR or common LGR which is basically the way we detect collision between an applied label on already existing or delegated labels. So we need a mechanism to determine that you can in fact create—the new label that is applied for is still available for delegation.

[The only one among those three sets, those allocatable labels in fact the Chinese LGR.] Those [have] variants. All in fact LGRs do contain variants, but in those LGRs, they are blocked. So just to be clear that it’s not true that the Japanese LGR does not have variants, that they're all blocked. We’re still waiting for the Japanese LGR to complete public comment, but we expect no major issue on that public comment. But time will tell. Next slide, please.

So that slide or this picture is also part of the root zone LGR overview, so you can see in fact more detail on it, but the important point here is that you have really two elements where you submit a label for delegation in the root zone. You have what we call the original label and then the proposed variant labels. So those are two paths how you get into the process. The first one is what we call the applied for label typically, so that's the one that is the main focus of your application on.

The applicant—obviously, we use element LGR to verify that the applied for label is valid, so it does satisfy repertoire requirement or rules requirement or any special requirement expressed into the LGR, and then we determine that either it will be rejected, it's not valid, so we stop there. Then we generate index variant which his basically a mechanism where we create a unique representation of that label that can be compared to existing labels that are already delegated or part of the root zone. If it is unique, that is, not already used, then we accept it. If it's not unique, we reject it. That means basically there's another variant of that label already delegated. So in that case, we basically reject it.

The second path is basically at the same time as an optional step, an applicant can also propose variant labels. The way you determine those is by your own kind of research, or you look in fact at the LGR to determine what would be a variant of the one you applied for. Then when you create those variant labels, you do obviously first have to make sure that it's valid, so you use element LGR again. Then we also have to make sure that it's allocatable. If it's not allocatable, it would be blocked and then it would be rejected as well If it is allocatable, then it becomes an allocatable variant and it could be delegated in the root zone.

There is an important point here, to be careful which one you pick as original label and variant label, because the mapping between variants is not symmetric. You may have a different mapping type between the original and the variant for each member of a variant set. So sometimes you have to be careful of picking the original label that would allow the variant label to also exist, because if you pick for example a variant label, sometimes the other ones, the former original label, could not be made allocatable. It's not exactly symmetric. So sometimes you have to be careful which one you pick first if you want to have both to be allocatable. You would have to look at the LGR definition and the variant set mapping, the mapping types used for those variant sets. Next slide, please.

So, this is basically—we have four scripts that have relation among themselves. That's Armenian, Cyrillic, Greek and Latin. Most of them are already complete. I think the Latin script LGR is the only one that is still in public comment. Greek went through public comment, through completion not long ago but it's not done. Armenian and Cyrillic were done a long time ago, so in fact [what we call deferred.]

So quick explanation how we do this, the variants inside that system are fully listed in each of the four LGRs. It's a bit different than what we did, by the way, for C/J/K, but ... So basically, variants that are related to the interdependence of those four scripts are part of each LGR. So you can basically read the four LGRs as kind of a consistent set in order ... If you find a variant in one LGR that is related to another one, you would find an equivalent in those LGRs.

We made an exception, what we call generic shapes. Those are basically like the O, the circle, or the C. Those in fact exist in other LGRs that have nothing to do with the Greek [inaudible] like the Armenian, Cyrillic, Greek and Latin. You may find them in some of the south Asian scripts that are already part of the root zone but they are not related in any way to those four scripts. So we decided in fact to, as integration mechanism, only list those in the Latin LGR when it is integrated. The other ones will obviously [inaudible] repertoire, but we didn't want them to be in the element LGR to cut the noise and not make some of those variant sets, otherwise it would be just huge in all cases and doesn’t really bring more information.

Again, as I said before, determining label collision requires use of a merged file we call common LGR. Validity is always done by each element LGR, that’s not changed. There was this deferred situation for two LGRs that was, that was Armenian and Cyrillic. Because we were not really sure until we had the four of them what would be the situation, especially when we put in-script variants, basically when one script imposed to another script, variants between two of the members of the script ... situation is quite common in fact. We have had that situation. For example, the Latin LGR did impose some in-script variants for Cyrillic that were not part of the Cyrillic proposal, original Cyrillic LGR. So we had to basically add those in-script variants to the Cyrillic LGR.

So that means that we see the deferred LGR had to have some technical changes, but we did talk to the generation panels or at least members of the [formal] generation panels to verify that that was acceptable. And those added mappings would be part of the integration, so that would be different from what original LGR were for Armenian and Cyrillic. But that was in fact negotiated with the GP members to make sure that it was acceptable. Obviously, the fully integrated set will be going through public review again as root zone LGR five. Next slide, please.

So the root zone LGR five content is the existing 18 scripts, part of root zone LGR four. We added two new scripts, two new C/J/K, because we already had Chinese before so we now added Japanese and Korean. And then we're adding two new alphabetic scripts, Latin and Greek. And like I said before, we have two deferred alphabetic scripts, that was Cyrillic and Armenian. And the last one that is still work in progress is Myanmar. We're hoping to get Myanmar [inaudible] in public comment shortly. I don't want to—that's more an ICANN message to make, not myself.

So the root zone LGR five is targeting 25 scripts. Two scripts are not done yet, Thaana and Tibetan. I've seen on the slide before it was mentioning 28. I'm not sure. I'm only counting 27, so there's one missing somewhere. Or maybe 28 was another slide. I don't know, but I only counted 27.

PITINAN KOOARMORNPATANA: Hi Michel. We have two for Japanese, katakana and hiragana.

MICHEL SUIGNARD: That's why. Okay. Yeah, because I was kind of wondering how come we have a different count. Anyway, so that's the conclusion. I have another slide but it's just for reference. So I'm done. I guess now is the time for questions if we have, for all of us. Thank you.

PITINAN KOOARMORNPATANA: Thank you, Michel. Sarmad, would you like to go back to the question earlier, or has that been addressed partially?

SARMAD HUSSAIN. Thank you, Pitinan. I'll read out the question. It may have been addressed by Michel to some extent, but we'll give Michel another chance to see if he wants to add anything else. The question is from Donna Austin. Has this conclusion—that is, no variants in—this was in the context of Japanese LGR—been reached in any other scripts? So that is a question. Has this conclusion, that is, no variants allowed, been reached in any other scripts?

MICHEL SUIGNARD: Again, it's not necessarily that—there are variants. Many other LGRs have variants. But most of them are blocked. Allocatable is a complicated situation, because if you allow allocatable variants, from a DNS point of view, it's not that simple. People more ... So we try as IP, we've been really making sure that if you define an allocatable scheme, you have to minimize the number of allocatable variants. For Chinese, I think it's a maximum of five with the original label that can exist. So if you apply for an original apple for label, you can only have four additional allocatable variants. That's the maximum.

So I think in Latin also, there is mitigation in place on LGRs to limit the number of allocatable. For most of them, we could achieve something reasonably simply. Japanese was really a big challenge for us because we worked very hard with the Japanese GP to find a solution to create the limited amount of allocatable variants for applied label. There is really no good solution. We looked and looked and we tried to work to identify a solution.

We cannot find a solution that we—because in the worst case, you can get a massive explosion, you could have thousands—if you create a bad label—bad, I mean a label that is basically just created to make it the worst case, you could have really—even if you have tools—the tools will not—there will be so many enumeration, it'll be so complicated that the tool could be running for days and not give you any solution.

So again, the situation was we could not have too many allocatable variants. So that’s why we came to the conclusion along with the Japanese GP that the only solution was in fact to block all variants. That’s how we got there. But again, they do exist, they're just all blocked.

It’s the current situation with many LGRs, by the way. Many LGRs blocked variants. It’s not something specific to the Japanese LGR.

PITINAN KOOARMORNPATANA: Thank you, Michel. Any additional question?

SARMAD HUSSAIN: There is one more question from Robert Nkambwe. “Is this work a subset of UA and vice versa? Sorry for asking this question, I'm still new to ICANN stuff.” So the question is whether root zone LGR is a subset of UA or vice versa.

And there were some responses in the chat, but Pitinan, if you’d like to take that.

PITINAN KOOARMORNPATANA: Sure. Thank you. Thank you for the question, Robert. Yes, it’s actually related. In a way, IDN is the mechanism to make the non-ASCII or the internationalized domain names—so, is the domain name in the local script or languages can be used technically, but then how to make the system or the website or application out there on the Internet accept these IDNs, that’s the UA.

Another part about UA is it’s not just about IDN as well. It’s kind of interconnecting. For UA, part of it is to make sure IDN is accepted in the application on the Internet, but also another [inaudible] which is the new gTLDs, the new long or new short TLD. You might have seen like [dot new stuff] besides the usual similar dots, now we have the longer dots, .photography or the new short one like [.run] and so on. So UA is covering both IDN and non-IDN domain names. So that’s a little bit interconnecting closely. So the UA and IDNs are going hand in hand. Does that address—

SARMAD HUSSAIN: Thank you. Yes. We actually have another comment, but I think it’s a good comment to discuss by Michel if possible. Also by Robert Nkambwe. So he's saying that, “I expected rigorous testing before integration into root zone. I haven't seen that in the flow.” So if Michel—we’re short on time, but if you can share some of the testing which is done on root zone LGR proposals before integration. Thank you.

MICHEL SUIGNARD: Yeah, one part that we ask every GP to provide, a set of test labels. We want that set of labels in fact to exercise the rules on the repertoire in the proposed LGR. We also verify that the existing TLDs of the existing root zone will get through the LGR. That would be kind of bad if they don’t. So we do validate the existing delegated root zone labels.

We also test the LGR against data, basically the existing—we go through very large copies of data for each script. You can find those online. We then verify that most of the words that we see in common use go through the LGR. And when we do find deviation, we've been going back to the GPs to ask why do we see such deviation. Sometimes it’s because the script itself has gone through some [inaudible] revision or whatever. There's some historic reason sometimes why that makes sense. And we have been satisfied every time that whatever we were getting as answer from the GPs was satisfactory.

So yeah, we do test them through multiple mechanisms. Those tests tend to happen before integration because integration is really mostly mechanical, basically getting the LGR in a shape that makes sense for the root zone LGR. But it's not really—the technical work is done well before that by the GPs, interaction between GPs and IPs.

PITINAN KOOARMORNPATANA: Thank you, Michel. I think [that's about to cover. We already exceeded] the time, so I will close the call here. Thank you, everybody, for joining and for the interaction. Please do take a look at the two public comment proceedings, and hope you can provide some feedback. With that, I would like to close the call, have a good rest of the day and have a good rest of the ICANN meeting. Hope to see you next week. Bye.

[END OF TRANSCRIPTION]