BARCELONA – ICANN GDD: IDN RZ-LGR Workshop
Wednesday, October 24, 2018 – 15:15 to 16:45 CEST
ICANN63 | Barcelona, Spain

SARMAD HUSSAIN: Hello. Who just joined online? If you're speaking, we can't hear you.

UNIDENTIFIED FEMALE: Hello, Audric. Is that you dialing in?

AUDRIC SCHILTKNECHT: Yes. Yes, it's me.

UNIDENTIFIED FEMALE: Okay, it's a little bit soft. If you can speak closer to the mic, that would be great. Thank you.

AUDRIC SCHILTKNECHT: Any better?

UNIDENTIFIED MALE: Hi. I can hear you better now, but I'm on the other line so I don't know how good they hear you [as being] in the room.

UNIDENTIFIED FEMALE: Audric, sorry. Can you try speaking again please?

AUDRIC SCHILTKNECHT: Yes. This is Audric speaking.

UNIDENTIFIED FEMALE: Audric, can you speak again? Sorry.

AUDRIC SCHILTKNECHT: Yes. Hello?

UNIDENTIFIED FEMALE: Okay, it is better. Thank you.

AUDRIC SCHILTKNECHT: I am speaking again.

UNIDENTIFIED FEMALE: Yes, this is okay now.

AUDRIC SCHILTKNECHT: Okay, thanks.

SARMAD HUSSAIN: Thank you all for joining the session today on Root Zone LGR. Today, we will actually go through a few presentations. Initially, we have Asmus Freytag on behalf of the Integration Panel discussing a topic basically some details around variants and how they should be defined within the context of Root Zone LGR.

Then we have Audric joining us remotely who is going to be presenting some more recent updates on the LGR tool. We just released a new version of that online, so he'll talk about some of the additional functions which have been added.

We also have Dennis Tanaka who is also joining us remotely. He's chairing the Root Zone LGR Study Group. This is a study group which is looking at the technical application or technical utilization of the Root Zone LGR, and he is going to talk about some of the work this study group is undertaking.

And then we have a couple of community updates from different generation panels. We will have Wang Wei who is going to present an update on Chinese Generation Panel and Dongman Lee who is going to present an update on Korean Generation Panel.

And then we'll end with a question and answer session.

So let's start with the first presentation, and I'll invite Asmus to take the mic.

ASMUS FREYTAG: My name is Asmus Freytag. I'm a member of the Integration Panel. This talk is about results of a bit of a rethinking of variants by the Integration Panel following the large number of LGR proposals that we received in 2018. Those have, as Mark mentioned in the morning session already, kept us very busy. But they also resulted in us taking a good look at making sure that we really handle things consistently and appropriately across the different scripts. And it caused us to look back

at the procedure as well, how it defines things and making sure that we are actually operating within the bounds of the procedure.

There's a number of topics that I'm going to touch. For those of you who have the online version of the presentation, the last two slides contain a number of useful references.

We went through the procedure and looked for the definition of variant. In Section A.3.2, the procedure says very clearly, "A variant label is considered the same in some measure by a given community of Internet users." So we're looking for something that is not merely close to but really substitutable.

The procedure contains a number of principles among which the Contextual Safety Principle is an important one. It says, "If a code point or any of its variants present unacceptable risks of being used in malicious ways, it should not be permitted."

The principles are all stated in terms of code points. I think they imply quite by extension also to labels. If any label or any of it's variant labels present unacceptable risks of being used in malicious ways. Often it is not so much that the label as such can be used maliciously, but if it were to be delegated and one of its variants were to be delegated to somebody else, that would open the door to malicious use.

When it comes to code points, because the LGRs of course are written in terms of code points even though what we're interested in is the behavior of the resulting labels, there are various different factors that can make code points substitutable. A really well-known example is the

Chinese case of same semantic. Chinese Simplified and Chinese Traditional ideographs have the same semantic even though they look quite different.

In the Ethiopic case which is in LGR 2, we had an interesting case of there being a series of code points that had effectively the same pronunciation with users tending to spell words based on the phonetic outcome rather than the identity of the letter. Here's an example of a character set [inaudible] all "ha" in pronunciation. This is an issue particularly in the more predominant language Amharic, and we had to apply that across the entire script even though it isn't an issue in some of the other languages.

Finally, we have the case what we call indistinguishable appearance. It can include absolutely the same appearance, such as the Latin letter and Cyrillic letter you see at the bottom of the list. But it can also include letters that if you don't see them in the context of some other letters, you would not be able to tell whether they are from a different script and therefore they would look to the user community the same.

And let me go back and make one more remark. These [inaudible] variants, when scripts are related, they tend to have a large number of overlaps. But some of them are of this kind of example of simple shapes which we think is highly security relevant and should be applied even if the scripts themselves are not directly related.

Now for each code point variant, an LGR would define a type. The type then results in a disposition of the associated label. If the only variants that exist in a label are of the type allocatable, then both the original

and the variant label may be delegated to the same entity. If at least one code point has a variant of the blocked type, either the original label or the variant may be delegated but never both.

These two types differ in the effect on security. The blocked variants very obviously prevent certain malicious registrations by preventing a second party from registering something that is indistinguishable. From that follows that the more blocked variants we have in an LGR, effectively the more circuitous within some limits of reason.

Allocatable variants, we would allow one entity to offer multiple equivalent labels where required but preventing registrations by unrelated entities. The downside of that is that it leads to a potential for combinatorial explosion. That is, at each level you can have multiple variants of some label and by the time you're done you have an explosion in the number of fully qualified domain names that are supposedly referring to "the same thing."

So one of the things we have done, we've worked with generation panels very hard to make sure that the number of allocatable variants computed by an LGR is preferably three or less. Or if that cannot be achieved strictly, that we try to achieve that for the multitude of cases.

Sometimes in the discussion the question comes up, "What about we're getting these extra blocked variants that don't really seem to fit and maybe we inherit them from the integration process from some other LGR that shares repertoire with our LGR. This is something that the procedure actually discusses and took into account. And it is very clear that even if an inherited variant is not necessarily linguistically

motivated, it is not necessarily a reason to reject it. Because the goal is not to maximize the number of possible labels but to minimize the confusion possible in a shared environment.

In effect, if you look at what the procedure says, blocked variants over and above some minimums set are definitely permitted, in fact, you could even argue encouraged. The procedure does not recognize the argument that imposing some blocked variants might reduce the overall namespace of variants. In fact, reducing the namespace is seen as conservative and the procedure has the mandate for the root zone LGR to be conservative. In doing so, the procedure focuses on making a shared zone, which is the root zone, safe for all users even if it as a result becomes slightly nonintuitive for some user community.

Over time, we've run into a number of cases where one community has inherited variants from another community. We have in Arabic the case of FEH and QAF which are normally distinct letters, but because of the existence of a third letter that they are similar to – I mean, identical to in some positional shapes – they become variants. More precisely, the third letter 06A7 is a semantic variant of the QAF and a visual variant of the FEH in middle and initial positions. Because of that, the two characters FEH and QAF even though they are normally distinct became variants. Doing so increases the overall security of the LGR even though normally one does not start the process with thinking of FEH and QAF as variants.

I've already mentioned the example of the Ethiopic homophones. The Amharic language, it's writing is characterized by the fact that there is

a wide variability of spelling. And apparently not in the sense of regional spellings like we know from English where color and colour can be spelled differently, but rather more flexible and individualized.

One could say in the extreme that users really spell the way things sound, not the way things look. And there are some deep reasons for it that you can find. If you want to look at LGR 2, you can look at the original proposal that explains it. It is based on the fact that the writing system was developed for a precursor language that made many, many more distinctions than modern Amharic does.

Even though they're not variants for other languages using Ethiopic, the generation panel proposed and the Integration Panel accepted that for security reasons these should be applied to all languages using the Ethiopic script. So all these other languages are now suddenly inheriting variants that are not necessarily linguistically motivated from within their languages.

You could think of that case as somewhat equivalent to the CJK case where you also have multiple languages sharing the same repertoire. And then we did some research and even though the number of homophones is quite large, we compared dictionary collections from some of the other languages and ran them through the LGR and we're trying to identify how many words in some of the other languages were blocked from each other.

We had expected to see numbers in the low tens of percent, when we actually did the analysis we found only 1% of the words and some non-Amharic language would actually because of the LGR collide with some

other words from the same language. That is a very encouragingly low number. And on top of that, one has to consider that the result of a blocked variant isn't that both words are unavailable. It means you set up a first-come, first-serve and the first word that is applied for would be the one that can be delegated. So from a point of view of an applicant in a process where nobody is ever guaranteed that they're the first to apply for something, this isn't all that big of a change we think.

The procedure is quite clear on the difference between the allocatable and the blocked variants. There's direct reference that having a strictly minimal set of variants that are allocatable would be beneficial. And when it comes to blocked variants, the procedure actually at one point mentions the aim to maximize the number of blocked variants. That is not to be taken too super literally. We don't want arbitrary random unmotivated extensions, but if a reasonable addition of variants causes a few additional or a few hundred additional blocked variants for large script, the procedure is quite clear thinking that is not an issue. It's, in fact, beneficial.

So to summarize, allocatable variants always have to be the minimal set that can be achieved. Different scripts make that harder and easier. Some of the CJK scripts have done really great work recently discovering very clever schemes to minimize the set of allocatable variants. And in return, when it comes to defining blocked variants, the procedure is much more open just thinking that more blocked variants really result in a safer root zone overall.

There is a generic issue with what we call the cross-repertoire variants. We use the term cross-repertoire here in this presentation instead of cross-script because some of the considerations apply in cases where LGRs share a script but not fully as it is the case in the [inaudible] script. After the generation panels have proposed their variants based on the inherent linguistic requirements of their users, the idea is that the integration process then applies these crosswise and any additional variants are introduced mechanically as part of the process to make the entire set transitive and symmetric.

The procedure considers those imposed variants as implicit and requires generation panels to provide explicit definitions and explicit dispositions of any inherited variants.

Now we need to be clear what that means. In principle, if there's a cross-script set or a cross-repertoire kind of variant, it could be the variant is defined from one member of one repertoire to another member of another repertoire. In this case, there is no need for any generation panel to make any more type definitions because once you have a variant that goes outside your repertoire, it cannot ever be allocatable so it has to be blocked. There's no choice.

However, there are cases where maybe Repertoire A has two or more variant definitions that go into Repertoire B and as a result you begin to pick up in-script variants inside the other repertoire. That is really critical because when it comes to in-repertoire variants, each generation panel must be able to decide whether they are blocked or allocatable.

The process is each LGR defines the variants based on their own requirements. It would inherit any applicable cross-repertoire variants from other LGRs. At least for the inherited in-repertoire variants, it must define dispositions. If those are not defined, the LGR would be rejected. It may optionally define matching cross-script variants. We've encouraged in some cases generation panels, especially for the European scripts, to go and have matching definitions because it's easier for reviewers and the number of variants are not very high. In the case of Korean, we have not insisted that the Korean LGR define the out-of-repertoire variants because that would really bloat their LGR proposal and make it harder to review. That's why there is the word optional in there.

Here's a couple of cross-repertoire variants. I'm not going to give CJK examples because they are so well-known. Here's a case that came up in recent discussion. The Cyrillic LGR, which is currently in a deferred state so that we can fully integrate it with Latin and Greek when those become ready, defines a variant between two letters – one Cyrillic on the left, one Latin on the right – that look like the letter "y." I wouldn't know what the name for it is in the Cyrillic script.

In doing their work, the Latin generation panel said, "Wait a minute. There's also another Cyrillic character used in Mongolian that looks a 'Y' with a straight leg. And we are really concerned that if there's no indication what script things are and you throw that 'Y' with a straight leg in there, an ordinary Latin user may well accept it because it's not unknown for some fonts to have shapes like that."

So if that were to be accepted, there would be an imposition of an in-script variant for Cyrillic by transitivity. And for that to be applied, the Cyrillic GP would have to define a matching mapping and assign a variant type. If that doesn't happen, then we can either not have this variant or not have the Cyrillic LGR.

As an aside, because non-IDN ACSII only TLDs have existed for a long time, it is never possible for any LGR to define things that would cause an in-repertoire variant in the ACSII set. No matter how reasonable it looks, it's something that the door has closed on that one. We cannot do that. It's a clear function of the process.

I mentioned at the start that we think of variants as things that are substitutable for each other because some part of the user community views the labels as the same. There is, of course, a number of cases where things become first slightly subjective, then more and more and more subjective. What the procedure aims at is that the LGR process should be designed to clear the table of all straightforward, non-subjective cases. In fact, every case that exists should be so straightforward and non-subjective that we would never expect any of the results of the LGR procedure to be appealed on an individual basis [on] an individual label.

Therefore, our considerations are limited as far as appearance is concerned to cases that are unambiguous, have overriding security concerns, and exhibit true exchangeability – are either homoglyphs which means a precisely matching shape or something that is

effectively indistinguishable even though with a magnifying glass you could find some tiny difference.

The presentation recently from the Sinhala generation panel had a number of cases where if you look at it this size across the room, you can stare your eyes out, you cannot see the difference. If you magnify the code point, you see a tiny little hook somewhere that is just almost invisible. So their conclusion is that users will see these things as the same in terms of identifiers, and we have proceeded on that basis.

There's one particular type of shape which I call a simple shape, either the straight line or a circle. And we have analyzed the circle and found it's very common across many scripts. Here is the list of circle glyphs that are standalone across the scripts selected for the Root Zone LGR. Because the glyph is so simple, if you see several of them in a row and nothing else, you have no idea what script it's from. You cannot tell. You can magnify to your heart's content. There is no clue.

And just to make matters more interesting, there is already an ACSII TLD delegated that is .ooo. Therefore, it is the position of the Integration Panel that a Root Zone LGR that does not treat circles as variants like that would be too insecure even though it means introducing a small number of cross-script variants between scripts that are normally not related.

If you look at the example list, you see the first four are absolutely identical. But even if you take the second from the bottom, Malayalam letter TTHA and if you have a .TTHATTHATTHA and maybe it followed a Chinese second-level label or some other script second-level label, you

couldn't be sure because Internet addresses just don't carry context. You couldn't be sure that wasn't meant to be the Latin .ooo.

This is going to be my last slide. We are sometimes asked there can't possibly be any Chinese visual variants because we take care of all the variants. But if you look at the CJK scripts, there are a number of cases where you have rather simple looking shapes that could easily lead to confusion. And the IP is very keen on having the generation panels for these scripts investigate these cases and handle them.

In the HAN case it includes a few cases. You see the second one from the bottom. It's a relatively complex character, and I would swear that on my system when I look at it the one on the left and on the right are absolutely indistinguishable. When I see it here, which is using a different font from the one I have on my system, you can see that they're not precisely identical. But is [clear] enough to think that there are common fonts as recent as Windows 7 that absolutely render these two identical. So from a security point of view that seems motivated to look at those cases and provide a positive disposition on them.

With that, I think we have covered quite a bit about the requirements for defining variants and the things that should be considered. We in the IP have certainly learned quite a bit in the process thanks to all the generation panels who have fed us with very well thought out proposals to consider. This presentation may serve as a very brief summary of the status quo of our thinking as it has evolved to date.

Thank you very much.

SARMAD HUSSAIN: Thank you. We'll take a couple of comments or questions. We have one online. We'll start with that and then we'll come back to this room.

UNIDENTIFIED FEMALE: We have a question from Bill [inaudible]. Question: "Are you more interested in blocking for security reasons variants whichever it uses will [find] indistinguishable versus those which are immediately distinguishable to professional linguists or Unicode gurus who spend lots of time looking closely at the glyphs?" And continue: "This becomes important in Latin where there are a lot of diacritics, but any given language [its] users only use a small subset."

ASMUS FREYTAG: This is a very interesting question. I would say that our standard is not that of a professional linguist and Unicode guru because a professional linguist and Unicode gurus are able to see all sorts of differences. Our standard is also largely not that of a naïve user but what we call the careful and observant user. Because we have many things that are similar enough that if you don't really look at a label, you may mistake it for something else and we are not interested in getting into that.

We are interested in where somebody is trying to do the right thing and is really unable to make a distinction because they are not a professional, because they don't have the tools or don't think of using them to magnify everything they're looking at, etc. So that's basically what we're looking for, the careful and observant user.

SARMAD HUSSAIN: Thank you. Mats?

MATS DUFBERG: I have a question about the picture here. On the first row, you have two glyphs on each side also with script mixing on the right side. So how come that you compare the combination of two and not just the glyphs that are very similar?

ASMUS FREYTAG: That is also a very good question. I think I have mentioned the reason for this in passing. First, these are examples. But ultimately we're interested in labels. Users interact with the labels and not with code points, so our examples here are taking possible labels. And what you think of as script mixing on the right would be permitted under some version of a Korean proposal. So all these under some versions of some proposals might be valid labels. Okay? That's the reason we constructed these cases.

Just as I showed you the example of .ooo of the Latin TLD to give you an idea that really you get labels that have three circles in a row. It's not that you're looking at the one code point. You're just seeing the whole thing. And the circles are a little bit bigger than they would be for a Latin "o" but if there's only three of them and nothing else in the fully qualified domain name as in Latin, you can't tell. You will think it's a Latin ooo. That's why we give you these examples of labels.

SARMAD HUSSAIN: Thank you. Let's move on.

UNIDENTIFIED FEMALE: [inaudible]

SARMAD HUSSAIN: Can we come back and maybe take it after the presentation? If it's a quick comment, maybe you want to.

UNIDENTIFIED MALE: Just one sentence. If you go back to the slide, basically the comment is that it seems to me that at least the first five of them don't really pass the non-subjective bar in terms of the differences. I can clearly see differences in all of the top. I'll leave the last one for another discussion, but that's my comment.

SARMAD HUSSAIN: Thank you. Let's move on. The next presentation is on the LGR Toolset. We have Audric joining us online, and he's going to be making the presentation in collaboration with Marc Blanchet who is here in the room. And he'll basically be talking about the new functional additions to the LGR Toolset which actually has been available online and also as open source for download for some time now. We just shared the new editions. Over to Audric.

AUDRIC SCHILTKNECHT: Thank you. Is the sound good enough?

SARMAD HUSSAIN: Audric, we can hear you, so please go ahead. And please let you know as you want to move the slides forward, and we'll do that for you from here.

AUDRIC SCHILTKNECHT: Okay, sure. Hello, everyone. I will be presenting the latest update of the LGR Toolset [inaudible]. The contents of this presentation will be as follows. [inaudible] will do a very brief summary of the tool itself, and then we will move on to the updates. Next slide, please.

The toolset is basically a tool to [inaudible] LGR [inaudible] using a user friendly interface with some [inaudible] checks that you don't have if you are editing your [XML] file with your [text editor] [inaudible]. We [inaudible] some tools. For example, to validate labels, generate variants, verify collisions.

The tool, as Sarmad was saying, is available open source, also online as a service. They are both the web interface which is a good interface for regular users as well as the [command] line and libraries code which are more for advanced users. Next slide, please.

[inaudible] to the tool was the addition of a harmonization process. The harmonization process will take LGRs and their input and will process them in such a way that they have the same variants mappings for shared code points between both LGRs. That means that they will have

for every code points that is presenting [with] two LGRs, they will have the same mappings, so the same variants. And we also respect the symmetry and transitivity. We also are able to discover variant mappings using a third LGR, most likely the Root Zone LGR. This discovery process is based on the code points script. Next slide, please.

We have also added some [inaudible] functionality. So, for example, now you are able to select multiple code points. This is step one. You can see that three code points have been selected. And then you have a new dropdown menu where you can choose an action to perform on these code points. For this case, it's to add the WLE to these code points. Then you have the popup showing. There you can enter the when or not-when rules based on what is already defined in the LGR. Then when you click on the "Next" button, they will be applied to all the selected code points. Next slide, please.

You can also do the same process for tags. Select the code points, choose "Add Tags," and then you can enter any kind of tags you want, multiple tags since they are space-separated, and they will be added to the selected code points. Next slide, please.

A new tab has been added to manage tags. In this list, you have all the existing tags defined on code points with a list of associated code points. You can use the list to, for example, if you want to remove a tag from the LGR, then you can just click on the garbage can on the right and it will remove the tag from every code point listed in the LGR that has this tag. So basically it's kind of a summary and deletion for tags. If you want to create new tags or edit tags on a code point, then you still

need to go [inaudible] using the tool that I've just presented, go edit code points one-by-one, for example. Next slide, please.

A new button has been added to populate variants. Basically, what that means is that we will ensure that symmetry and transitivity is [respected] on [inaudible]. So we will add missing variants. We will add missing mappings and be sure that the resulting LGR is symmetric and transitive in respect of the variants. Next slide, please.

We have also added some kind of very simple tool which is not really related to an LGR document itself but is available again as a helper tool. The tool will display all the forms of the label. So you can enter a label, choose the Unicode version, and then the tool will display the three forms of the label, meaning the code point sequence, the U-label, and the A-label. Next slide, please.

We have been improving the Add Variant button function, especially in the case when you want to add out of repertoire variant. What that means is that now we ensure that the variant will be added to the repertoire and that [reflexivity and transitivity] are still ensured on the resulting LGR.

For example, if add the variant 0E20 and then click on the button – next slide please – you will see that in the box number two that the code point was added to the LGR as well as – sorry, the code point for the variant was added to the repertoire as well as the variant added to the edited code point. So the variant is visible in the square on the box number three. Next slide, please.

Also, now when you want to add code points to an existing LGR or some new LGR that you are creating, you can select code points from a script. You choose a validating repertoire, for example, [inaudible] you choose the script you want and then you have a list of add code points for the script defining the validating repertoire. So you can select which code point you want, and then they will be added to the LGR. Next slide, please.

This slide is a bit more for the behind the scenes updates. First of all, we have worked on performances, especially for dealing with very large LGRs such as the CJK one. So now we are able to do that in a timely manner. Before it was very, very long just to load the LGR. So now you can actually edit and work on a very large LGR.

We have also added full Python3 support because Python3 is the way forward as the programming language version. So especially as the tool is open sourced, we need to support the latest version of the language. So that was one of the big improvements of this release.

Some work has been done on being more explicit about failing rules. When, for example, you are trying to validate a label, now we try to be a bit more precise on what code point or what variant is making the label fail validation.

Support for Unicode 5.2.0 to 10.0.0 has been added as well as support for MSR-3. We also display now the combined form of sequences which was not [inaudible] before. Before, sequences were only shown to the user as a list of code points, and now we also display their combined form. This is done both in the repertoire view, so the main view of the

tool, as well as in the HTML output. Obviously, we also did some bug fixes on the tool. Next slide, please.

That concludes my presentation. You can see the tool online at the web address listed on the presentation. [All] the tools and libraries used are also available at GitHub released on the BSD license. There is also some information and user manual posted on the ICANN website.

Thank you.

SARMAD HUSSAIN: Thank you, Audric. Any questions? Yes, we have a question. [inaudible], please. Raed, please.

RAED ALFAYEZ: Yes, this is Raed Alfayez from SaudiNIC. First of all, I would like to thank the developers and ICANN for providing this very good tool. We were using it maybe for more than one year ago. One of the things that I have noticed recently and it makes sense now what is the reason behind this, it was the feature for automatic addition for code points and variants. This contradicts with the our language table because sometimes we define the code points for the Arabic language, which we all know about them, and sometimes we need to add a variant that is not part of the Arabic language. Maybe part of the Urdu or for the Farsi code points.

Once we capture or we add this variant, so I will take the example of [inaudible]. We have the Arabic [inaudible] in our code points as [an essential] character in the Arabic language, and we need to be able to

the Farsi [inaudible] so people in Pakistan, Afghanistan, all of these countries can reach our domain name. And this variant is allocatable and most probably it will be activated.

The problem is that now I cannot continue with your language table unless I add a [inaudible] tag for the Farsi [inaudible], and this is a problem actually. So I cannot have a character not from the Arabic language to be part of the Arabic repository, and this is a problem. So I believe you reconsider this feature or at least solve it in a manner so that I shouldn't be forced to add a character in our language that is not part of the language itself. Thank you very much.

UNIDENTIFIED MALE: An RFC problem. May I? I think you'll be disappointed by the answer because it's a result of the way the underlying XML format is defined in RFC 7940. It is not intended to capture a definition of a language, but it is intended to capture the permissible list of code points for labels. So if you have an allocatable variant that therefore may become part of a label, it is by that fact a part of the repertoire of that zone. What you can do if you like to document which ones are part of the language and which ones are not, you can use either references, comments, or tag values to identify proper language members. But from the point of view of label generation rules, anything that can be part of an allocated label is definitely part of the repertoire of that zone.

RAED ALFAYEZ: I agree with you, but the problem that someone reads this table, he will consider that the Farsi is part of the Arabic language and this is difficult unless we have some kind of tag like not [inaudible] tag. Like something transitivity tag that we can use just to differentiate at a very clear level for anyone who reads or implements this LGR.

UNIDENTIFIED MALE: I would suggest that most end users who might be confused by this are people who are much more likely to read comments than they are looking for other types of information. So if you provide a comment for that code point saying, "This is a Farsi character available in the zone even though it's not Arabic," I think you have made your point. Also, don't forget as you publish an LGR it has a large description section at the top where you can very carefully discuss that case upfront and then people can read it in plain language what your intention is and what you're doing and what the result is. So that's how I would go and address these issue.

RAED ALFAYEZ: Yes, but still there might be systems to deal with the LGR, so not only programmers. So a programmer will develop something, then system will take it, include it, and then display it in a registry system or somewhere else. I think at least even in the RFC if you go to the [inaudible] tag that defined the RFC, it says this is part of the [inaudible]. But this is actually not the case in our case. This is not part of it. So I don't think just description or maybe put a condition when just only blocked every time, I'm afraid this will not give the good

inception for what the [inaudible] tag is used for. I hope that maybe consider it or have another tag that explicitly shows this is not part of the language table for this LGR and this just has been added just for transitivity purposes. Thank you.

SARMAD HUSSAIN: So I think your point is well taken. From the tool perspective, I think this discussion has gone more beyond the tool's perspective. [I think] in the next release we'll look at the possibility of not forcing anything like that and let the user, for example, have some flexibility in how they want to define an LGR. So we'll look, in the next release when we come to that, we will look at how we can make the tool more flexible.

We'll take one more comment from Edmon and then we'll go on to the next presentation.

EDMON CHUNG: On that topic actually from .asia, we were just starting to look at the Arabic [and suite] of issues. I think one of the things that came up immediately is that we would have multiple language tags and multiple LGRs. What you can do, what we are doing, would be create a WLE. If you create a WLE that says that string contains those characters, then you implicitly have that tag that you want in the LGR. I would say you can achieve that with a WLE.

SARMAD HUSSAIN: Let's move on. I think we're going into LGR designing here, which is beyond the scope of the tool itself. Thank you very much, Audric and Marc.

We're now going to go on to the next presentation on the Root Zone LGR Study Group. We have Dennis Tanaka joining us online to present, and we have some members here in the room as well. So over to you, Dennis. And for changing slides, please let us know and we'll move the slides on.

DENNIS TANAKA: Thank you so much. Just doing a mic check. One, two, three, four. Can you hear me?

SARMAD HUSSAIN: Yes, we can hear you. Please go ahead.

DENNIS TANAKA: Thank you, Sarmad. Next slide. We are going to talk about the study group on the application of the Root Zone LGR. The agenda will go through background, scope or work, current status, and next steps. Next slide, please.

As a way of context, we know many of you are familiar with the Root Zone LGR, now on Version 2. We heard this morning that there's an upcoming Version 3 on Q1 next year, so several scripts already integrated and many others in progress.

The study group was formed to look at doing a technical assessment of the implementation or application of the Root Zone LGR when talking about existing TLDs and the next round of top-level domain names, those being either ccTLDs or generic TLDs. What we hope to provide the community is those technical considerations that should be taken into account when defining subsequent policy for applying the Root Zone LGR in [reviewing] top-level domain names for the root zone, that is. Next slide.

The composition of the study group is right here. We have nine members. We try to hold weekly calls, and we're moving along. Next slide.

Now we're jumping into what we are dealing with. Some weeks ago, we closed a comment period on the scope items. The first task of our study group was to define what's the scope of work. We know that the [constraint is] only look at the technical issues, but then again we have to go in more detail and define what we have to deal with. So for the purpose of this presentation, we put this into these buckets, into the WHO/WHAT/WHY/WHERE/WHEN framework and also an Other Considerations box.

WHO will use it? It's obvious that the primary user may be a top-level domain name applicant, either country codes or generic TLD for that matter. Also, other users, generation and integration panels and other stakeholders such as ICANN organization or the PDP agents. Each of these users may have different use cases, so through the lenses of

technical aspects we will provide some recommendations as to what type of issues or considerations they need to be aware of.

WHAT does it do? The LGR? At this point, I think we're clear that the LGR serves two main functions which is syntax validation as far as which code points are valid for a TLD label and also the calculation and determination of the disposition values of the variant labels. And then we [get again], what if the Root Zone LGR calculation is not accepted? I think that was one of the questions that we were asked to look at if a TLD applicant does not agree and wants to appeal to the calculation. So what needs to be taken into account for that matter?

On the third bucket we will try to answer the question: WHY is it important? I think at this point, we established that we need a single source to validate TLDs for consistency and predictable results. However, what if the script is not supported in the LGR? For example, if we take the LGR as is today, there are many scripts not yet integrated, for example, Latin, Cyrillic, [inaudible], Hangul, and Japanese. What if there is a new, hypothetically speaking, there is a TLD application window and I apply for a TLD whose script is not yet supported with the LGR, what do I do? What does the application process have to do or can do to process that label? Is it going to be put on a suspension, or will it go through a separate process? So those are the things that we are looking at.

WHEN do you apply it? Not when do you apply it in the application process, but we're looking more on this LGR tool, I think it came up this morning also the question, why is it important that we have an LGR

tool? It's because there are existing TLD labels that will have variants, and so we want a consistent [inaudible] way to calculate those variants and their disposition values, whether they are blocked or allocatable. Of course, for the next round of gTLDs many of you are already aware there is a subsequent procedures PDP looking at the next policy needed to open up the next application window for TLDs. Also, we hope that the technical considerations are taken by the ccTLDs as well in their fast track process, which unlike the gTLDs fast track process works on a rolling basis meaning that it's a first-come, first-serve application process. We also want to or there is a need to apply the LGR to reserve TLD labels so that we know for sure what are the variant labels. And if there are variant labels, those need to be reserved as well.

Moving on to the next bucket: WHERE do you find it? The LGR is an [inaudible], it's an XML, right? That's what it's produced. The process is – you already know this, just to repeat it – generation panels provide the proposals. Integration Panel then integrates and gives us the LGR. And then the XML is [a] normative and there's also supporting documentation. Where can we find this authoritative documentation? We're talking about what is going to be the repository. Some of the options that we've talked about are IANA. IANA, you know, is the repository of all the IDN tables for top-level domain names, so it seems like a natural repository for the Root Zone LGR which is the master IDN table for the root zone, if you will. So it looks like a natural place where the normative XML and the supporting documents should live and maintain. But IANA is the repository. Who is going to maintain it? Is it

going to be [ICANN] organization? Is it going to be a third party? Those are the things that we are looking at.

And then lastly, Other Considerations that do not fall naturally in the previous five buckets. We'll look at the different variant states and how a label can transition from one to another, if that's even the case. We're talking about how an allocatable label moves on to the activated state, whether activated state can go back to an allocatable, meaning non-delegated. We're going to look at those items and what are the technical aspects that need to be dealt with.

Also, other security and stability considerations the [DNS panel review] need to asses. As an example, single-character IDN TLDs. I know there is some conversation to allow the delegation of single-character IDN TLDs in [certain] scripts. So what are the considerations that need to thought about? Next slide, please.

That was what is in scope. Here we want to clearly state what is not in scope, mainly because some items we receive through the comment period and some other things that the study group itself discussed. We are not to deal with semantic validation, meaning what constitutes or what is eligible for an IDN ccTLD, what's the criteria to process a label as a geo-name or brand or community top-level domain name. We clearly said those are policy work that regardless of the label being valid or not valid that has to be processed like a subsequent process which IDN ccTLDs have eligibility criteria for fast track and the SubPro will deal with geo-names, brands, community, etc.

Also, as far as limiting the number of allocatable variant TLDs, since this group does not challenge what the Root Zone LGR produces, we are not also challenging the number of allocatable variants the Root Zone LGR will produce. But we will have a comment on and likely will echo what's been discussed so far as having large number of allocatable variants for a single label. But policy will have to decide how to deal with those.

Also, coming back to the scripts that are not supported in the LGR today or in the future version, what's not in scope is how to deal with those. We will certainly look at the technical considerations that subsequent policy will need to look at when a script is not supported, but we will not dictate or issue recommendations as to how they should process that. We'll just give a technical assessment of what they need to look at, be mindful of, but the how is for their work. Next slide, please.

As far as status, we are still going through the analysis of the scope and drafting preliminary recommendations. Tentatively, we want to have a draft final recommendation version by November in the mid-November timeframe, which will lead us to a comment period in December and then have a final document or report by early next year. Next slide.

We end up with resources. If you want to, all our calls are recorded and minutes are published in the wiki page, and also you can look at our mailing list in the link below.

With that, that's it. Happy to take any questions.

SARMAD HUSSAIN: Thank you, Dennis. Are there any questions? We'll take one question due to the shortage of time and then move on. Yeah?

DONGMAN LEE: Hello. This is Dongman Lee from KGP. I have a very naïve question. I just want to know what kind of benefit, for example, the Korean community gets from this activity. What I'm asking is I would like to leverage or get some help from whatever they are trying to do, but because of my limited knowledge I don't actually grasp the key, the benefit. I'm not saying they're doing a bad job, so don't get me wrong.

SARMAD HUSSAIN: Dennis, do you want to respond to that? Or somebody else in the room who is a member of the study group?

DENNIS TANAKA: Sure, I can.

SARMAD HUSSAIN: Go ahead, Dennis.

DENNIS TANAKA: Thank you, Sarmad. I think we have to step back. That's a good question. Why is that? This is the way I look at it, so please if other people have different input, please chime in as well.

The way I will look at it and the way it was presented to me is that the Root Zone LGR today, yeah, we have a tool, we have an XML that everybody can use. But it is not yet integrated into the policy. So if there is a subsequent round of gTLDs, they are not mandated or bound to use the Root Zone LGR in order to validate TLDs. So this is a precursor or a [pre-step] for policy work to adopt the Root Zone LGR by looking at the technical aspects that they need to take into account in order to adopt it and integrate it into the next – just using gTLDs as an example – in the next version of the Applicant Guidebook. Something similar to the fast track process.

SARMAD HUSSAIN: Okay, thank you, Dennis. We'll move forward. Next we have an update from the generation panels. We have two updates from the Chinese Generation Panel as well as the Korean Generation Panel. So first, I'll request the Chinese Generation Panel chairs Kenny Huang and Wei Wang to present an update on the Chinese Generation Panel. So over to you.

WEI WANG: Thank you. Good afternoon, everyone. Here is a brief introduction about the CGP updates. We have 23 members from 10 countries in the region. We have the advisor appointed by ICANN, Edmon Chung, which will help us to coordinate within CJK and help us coordinate between [inaudible] and ICANN.

The latest version we provided to ICANN is Version 11 based on the feedback from an IP in February. We updated the document in August. I have to remind everyone that the Chinese character is not only used in Chinese language region but also in the Korean and Japan community. So for which we have a list of over 4,000 mapping characters between C, J, and K.

The biggest change in the new version is that we removed some characters imported from the J and K in the last version which makes decreased the number of the repertoire. It removed about 140 characters.

Also, besides the repertoire, we coordinate with J and K for some other [arguable] variants. In 2015, the Korean community raised about 445 variant groups that [could not accepted] by the Korean community. We conducted the pre coordination and integration between the C and K and reached a consensus in 2017. That is what I mentioned. In the new version we removed those characters imported from J and K and from the normalized [Han] character list published [inaudible] in 2015 which decreased the number of repertoire to 19,685.

Because we removed the characters imported from J or K, consequently we [developed] a new subtype of variant which is out-of-repertoire variant. We noticed that in the meeting in Puerto Rico, IP [raised up] the visual similarity issue. And we analyzed this issue and we have to admit that there are some Chinese characters, [Han] characters even for the Chinese users in some circumstances it is hard for them to tell the difference to distinguish them from each other. For the Chinese

characters there are Chinese-Chinese pairs and also there were some [Kanji] and Chinese character pairs and also there are Korean, Hangul, and Hanja pairs. The CGP, we are willing to conduct the research analysis on it, and if the local linguistic experts think they are visually similar, we would accept [them] as the official identicals. Also if J or K generated their own very similar pairs, the CGP would adopt those similarity mappings.

For the next steps, according to the feedback from IP to the latest version in August, we the C will provide further detailed information about the C and K coordination to ensure that the current coordination will not split the traditional variant groups and will not bring any risk for the end user to misunderstand the variant relationships.

The second work is to generate the visual similarity list, as I mentioned. There are about four or five visually similar pairs that IP suggests the CGP to make further investigation.

The last one is the further interaction with the IP. We will have an interaction meeting with IP tomorrow morning, so we will have further discussion with linguistics and from IP and from the Korean community.

That's all.

SARMAD HUSSAIN: Thank you, Wang Wei. Let's move on to the next update from the Korean Generation Panel. We have Professor Dongman Lee who is going to be presenting the update on the work by KGP.

DONGMAN LEE: Thank you for the introduction, Sarmad. My name is Dongman Lee. I'm from the KGP. I'm presenting this one on behalf of the KGP chair, Professor Kyongsok Kim. He's very sorry not for coming this one. He has the business conflict.

Anyway, let me just go quickly and more focus on what we have actually done last six months based on the public comment. As you may know, the Korea probably among CJK is the first one who officially submitted the LGR proposal. So I'll just walk through quickly the overview of the Korean language and the members and what we have done so far and the next plan.

As you may know, the Korean script includes not only Hangul but also Hanja. We have over 2,000 years history of borrowing Hanja from China and then used in our daily lives over the history.

We have about 18 generation panel members across the technical through the registration agency.

The proposal we actually submitted last year in December, the Version 1.0, include the Hangul and probably there's no [argument] over there and Hanja repertoire. We actually worked with the Chinese and Japanese GP members and we came up with [already agreed] variant set. Wang Wei already explained, so I don't need to explain over.

This is actually very hot part. After we submitted the proposal, there were a bunch of public comments. The summary of the public comments it is somewhat kind of our interpretation. Maybe if you walk

through each single public comment, let me just be more precise. Some of the public comments actually explicitly opposed the inclusion of the Hanja. How that actually [inaudible] throughout our local meetings – and let me just explain in a few minutes.

Another key message from public comment was some people actually raised a concern about the usage of the Hanja in Korean language are not precisely explained in the document. They believe that as far as the Version 1.0 is concerned the [description] is too much overemphasis on the usage of the Hanja. But it depends on how you look at it.

Anyway, we started having the public meeting with the people who actually made such public comments over Skype and face-to-face meetings. Anyway, one of the things that we found that they're not really opposed to including Hanja. So Hangul-Hanja, they agree that both are part of the Korean language. Because of time, let me just – what we have not agreed so far is inclusion of Hangul plus Hanja mixed labels.

Here there is a person named, Mr. Byeon, one of the people who actually made public comment. He generated an about 20-something list of items for suggesting revision or correction or some changes in the proposal. And we actually carefully and precisely walked through all the items and then we resolved I think all of them.

On Monday, we had a meeting with the IP and we decided to include the English version of the [inaudible] how we actually revised our document as an addendum in the next round of the proposal. This one

is all that I mentioned. Still we have some disagreement on whether Hangul-Hanja mixed labels really should be included.

Let me just explain why people have two different ideas. As far as I'm concerned, I think this is just simply which education on Hanja in certain time you are educated. For example, me and my colleague Mr. [Chung], we were asked to learn Hanja through the education system. But another person, Mr. [inaudible], is about 20 years younger than me. He was not exposed to the Hanja education throughout his school days.

So what seems to me, I found it throughout this meeting very interesting really. People at his generation felt Chinese characters are more foreign than English characters. So when Chinese characters are introduced, the people in that generation felt like it is very foreign. So this is not just technical issue, more on the cultural understanding on whether Korean language should include Hanja with the Chinese characters or not. So it depends on which education system background you actually came from, the perspective is completely opposite. So we'll continue to actually try to have some consensus until the proposal submission.

So that's about it. Thank you.

SARMAD HUSSAIN: Thank you very much, Professor Lee. This brings the presentations to an end. We have a few minutes before we close the session, so let's open the floor for any comments or questions on these two presentations/updates by the generation panels or more generally on

the material which has been presented so far. So let's open the floor for questions.

Asmus, I remember you wanted to respond back to what Edmon I think had said. Do you still want to make a comment at this time?

ASMUS FREYTAG: Well, there's a big and a small comment. The big comment is the evaluation belongs into a generation panel, and when the generation panel has spoken and given a rationale for their decision the IP will look at that and work on the basis of what is on the written record. That is always the case.

In evaluating certain things, I just would like to throw out a small detail. We don't think that the best guidance to things is to have two different labels printed next to each other and then see whether they look different. I think we need to go back to the concern that underlies the entire procedure is the complete absence of linguistic and regional cues in a single fully qualified domain name. Especially on the root levels, you do not have the luxury of knowing a priori what to expect. So when evaluating a generation panel's decision making, we would probably look for that kind of understanding shows through in the written record.

SARMAD HUSSAIN: Any more comments, questions? Great. Then thank you all for attending. Let's also thank the panelists and speakers for their presentations, and let's close the session. Thank you.

[END OF TRANSCRIPTION]