ICANN79 | CF – New gTLD Program Next Round: Planning for String Similarity Review Work Session
Tuesday, March 5, 2024 – 3:00 to 4:00 SJU

PITINAN KOOARMORNPATANA: Hello and welcome to the Planning for String Similarity Review Work Session. My name is Pitinan and I am the remote participation manager for this session. Please note that this session is being recorded and is governed by the ICANN Expected Standards of Behavior. During this session, questions and comments will only be read aloud if submitted within the Q&A pod. I will read them aloud during the time set for question sections. If you would like to ask your question or make your comment verbally, please raise your hand. When called upon, you will be given permission to unmute your microphone. Interpretations for this session will include English, French, Spanish. Please state your name for the record and the language you will speak if you are speaking a language other than English. Please speak at a reasonable pace. With that, I will hand the floor over to Samad Hussain.

SARMAD HUSSAIN: Thank you, Pitinan. Good morning, good afternoon, good evening everyone here and online. So this particular session, we have been working on developing string similarity guidelines. What we will do is just give you a brief background. I am going to skip the contents slides.

So basically this is work which is part of the next round of gTLD applications as part of the policy development process. The initial evaluation of TLD strings will undergo a string similarity review by a

string similarity review panel. The IDN expedited policy development process phase 1 recommendations actually ask specifically to develop string similarity review guidelines so that the string similarity review process can be conducted more transparently for the applicants and the community.

So what we are doing as part of the implementation of these guidelines from SubPro and also with the guidance being provided by IDN-EPDP phase 1 is starting to develop these guidelines. There are two challenges we are trying to resolve. One of course is that we have to determine a reasonably clear or transparent mechanism to determine what string similarity is for everybody to understand.

And then the second challenge which we are actually trying to solve here is that with the advent of variant TLDs, the space in which string similarity is to be performed has increased or broadened considerably and because of that it may actually not be feasible to perform a complete manual review. So what we are proposing is that there is a pre-screening process which is done more mechanically based on some data which we will collect through communities and then based on that what that would eventually lead is the string similarity review panel goes through a pre-screening process and we have a smaller set of actual labels which can be then manually reviewed.

So what we have done is this is the first phase of these guidelines. These are now published for public comment. They are currently open for public comment. The links are available in the presentation and it presents the factors which determine similarity which can be used to address the transparency part of it but also the two-step process which

I just mentioned to address the scale of this work more efficiently. So we have actually been working with a team which is developing these guidelines and I will now hand it over to Michel Suignard who is a Unicode Fellow. He has been working with us to develop these guidelines along with [inaudible]. And over to you to take us through these guidelines and then please feel free to, if you have any questions, feel free to raise your hand and we can stop and take them at a logical place.

MICHAEL SUIGNARD: Yes, I also want to mention I have been part of the integration panel on the ICANN side, so basically working on the root zone LGR to help the various LGR coming from the generation panels to be integrated on what is now the root zone LGR. So I was part of that small group of experts. So I have basically some experience on what was variants, for example, on how to deal with multiscript context for the root zone.

Yes, so variants are very important on this study and we see that during the presentation and obviously we assume that the first thing when new application or new applied layers will be presented, we don't want obviously variants to be presented because they will be eliminated as the first part unless they are provided to the same entity, I guess. And like I said, it is a two-stage process. I am not going to repeat what we just said. Next slide, please.

Yes, I mean obviously we will go through some concepts, so we have to establish our terminology, you know, about what is a confusability, similarities and variants. I have some examples here. You see, for example, what we see as the same. Like you see something that looks

like a C, but is also a Cyrillic S. In the Arabic you have a [inaudible]. So those are really identical in shape, so those are eliminated in the root zone, for example, as being variants. And visually similar on the next row, you see a [inaudible] or in fact a final sigma in Greek. Then you see two examples in Devanagari, I think those are the [inaudible]. And those are in fact only visually similar. They are not variants in this situation. The last nine I will show, you see things that are totally visually distinct, so there is no point in analyzing those as being similar. Next slide, please.

So first thing, confusability is the degree to which two labels can be seen as being different and user confused. That is a confusion issue. When you determine confusables, we use confusable as both a name and an adjective, so don't be surprised. It is when you have two labels where the degree of confusability exceeds the defined threshold. The threshold is important because it is not really black and white. You decide what is the threshold based on your purpose, which is in this case the root zone LGR. We are not trying to make confusable as a general consideration. We apply it to the context here.

We have some constraint in the situation that some of those strings can be extremely short. In root zone, it is not uncommon to have two letters, the string, or even one. It is possible we could see some of them with one. Let's say we take the Chinese, we could see a situation where there is a single character. Sometimes they look like a single syllable. It is kind of a variable. It does include ASCII domains. That is a very different thing that we did for the root zone LGR. We did not consider the ASCII domains. We did not consider variants among ASCII because that was not possible.

Here we see similarity exposed in things that you don't see typically too much in the root zone. The difference in variants is that you may have letters that change depending on the font style you apply to them. What is a Cyrillic I that does not look like anything in Latin, suddenly when you apply Italic to it, it looks like a U. On the small F, which is when you apply again Italic to it, it looks like an F with a hook. In fact, the Latin LGR in the root zone did eliminate those two as variants. In some cases, you see there is some boundary sometimes between what is a variant on a visually similar gray zone where sometimes some generation pattern on the root zone decided to make some of those you could consider visually similar to be in fact variants.

Variants are a bit different in that situation. It is kind of interesting to establish what a variant is compared to vision similar. Basically in variants, you estimate that the variant label is in fact the same. It is not limited to visual similarity or being the same on the visual point of view. We allow that to be between single or multiple letters across scripts. You can apply to visual appearance, like I said, but also you can apply to meaning or semantic and also pronunciation. The most typical case of semantic is Chinese. We do create variants between simplified and traditional. In some other scripts, like Ethiopic, we apply phonetic, the sound to establish variants. We don't do that in similarity. Similarity is strictly visual.

Because variants are equivalents, we have symmetry on transitiveness. It basically means that if you have, for example, a triplet ABC in a variant set, if A is equivalent to B, B is equivalent to A. Then ABC, all of them play an equal term between A and B and B and C on A and C.

The variant disposition is always blocked. That basically means that the variant labels are not allowed if the original was used. Or they could be allocatable in some case. That means you can have multiple delegations of basically the original label plus the variant labels. That is following strict rules, again on the root zone LGR, to limit the number of those situations. That applies mostly to Chinese and Arabic, in fact, which are the more typical cases of this delegate allocatable situation. That's why we also use a term of original label or primary label, which is basically the one you applied for. You may have also [inaudible]. Then you consider the variant label. In a lot of things we do in these guidelines, there is really not much differences between the original label and the variant labels or the similar labels. They're going to all have to be examined in the studies. Because in fact, there is no meaningful differences between the original label and the variants. For example, here you see an example that was showing before between the small f and the f with hook, which is obviously a variant of each other. Next slide.

Yeah, we see when we do the similarity review, we have to look at the recognition task basically on context dependence. Those are going to be important factors. Because you can't again evaluate similarity in a vacuum, you have to consider context. Next slide, please.

So first we have to see what is a user. Depending on what you ask, you will find you have very different answers to what is confusable. We obviously assume that if you're looking at a script, you have to be a native user of the script or at least be familiar with it. You also need to have some concept of familiarity of the ASCII subset. Why is because the root zone is mostly in the very vast majority as ASCII in it. So you cannot

ignore the fact that 80 percent, 90 percent of the root zone is today in using ASCII strings.

So then there's also the context of how you recognize the two labels or the labels to compare. They don't need to be present at the same time. They don't need to be on the same location or the same -- you just need to be familiar with one. And basically in another context, it looks like the one you're familiar with. It does address, obviously, cross script on cross languages. I mean, when we say cross languages, we mean cross languages used by the same script. We don't expect to be able to recognize similarity on the script you don't know, you're not familiar with. It's not a requirement. I mean, if you know, somebody from Europe is not asked to recognize differences in the Chinese text, it's a whole label thing. It's not character by character. We always have to assume that recognition is done at the string level, not code point by code point. Next slide.

[inaudible] on the same point, where similarity typically involves what we call a perception space, where it's not -- they don't really live on the same location if you look at what looks the same. And for example, a clear case is when you have, again, a triplet of similar characters. Two of them may look like a third one, but they may not look like the one in the middle.

Again, the ones in the middle look like the two of the extreme, but the two on the extreme don't necessarily look visually similar. We have that situation. So that's what we call a perception space. And again, variants are not limited to visual similarity, which creates its own challenge, like I said before. And we see variants don't consider at all uppercase

situation. And that is now in visual similarity, we have to consider that situation. It does create some challenges.

And obviously, variants are not enough to do visual similarity. I mean, they were not meant to do that. They were meant to detect basically sameness, not, you know, similarity. I mean, there's been some debate. It's not like -- there's again a bit of a gray area, but at least the variant are for sure not trying to make an answer to all similar situations. Next slide.

Yeah, so that's where we see a way to run this, that we're going to create something similar to what we call a variant set. A variant set is basically a set that includes all the variants, code points or strings that are considered the same. And we use that in the LGR system to create those rules on established disposition for labels. It's basically a mechanism where you have a set and also you have mapping between each code point or sequences to determine what to do with them. You know, you would have to go into the LGR system to have more details. I don't have time to go through all these things here.

On the confusable cluster, it's basically the same thing, you know, applied to visual similarity. We basically create a cluster of all the confusables, and we group them as they were variants. So then you kind of create, again, a transitive situation. Obviously, the issue is that we're going to create a bit too many of confusables. You know, you're going to have overproduction of confusables by doing that to a degree. So that's where we have to be careful. There's a bit of experimentation to do there to get results that could be used, you know, in this, again, the pre-screening and post-screening mechanism. But the idea is that we

want to use, yeah, the variant system, again, to help us here. Okay, next slide.

Yeah, interaction with string similarity with variants, again, has to do with that when you compare to the delegated set, you would have to remember that you also have to consider the variants of those delegated sets when you do the comparison. And the same thing for the candidate labels as also his own labels. So you get a many-to-many thing. That's why, as Sam had said earlier, it's basically impossible to solve as a manual review. There are just too many of them. And variants can be almost infinite. You know, a label, especially if it's a long label, can have almost infinite number of variants in some situation. It's pretty mind-boggling how many variants can be created with some existing labels today. So you cannot solve that by just looking at them. Even today, we're only like, you know, 1,600 root zone TLDs. And even that is already too many to do a manual review. Next slide.

So we have a situation where we're mixing confusable and variants. And that's a small example. I mean, the first word is a French word for plane, aeronef. And then we kind of map it to ASCII. And we just got rid of the ornament, the acute on the E. So that's -- it's in fact -- it's not a variant in this case. It's a similar character because the E acute and the E are not variants of each other, at least in the hood zone.

And then your situation at the bottom where you have -- you're mixing now a variant because F on -- F with hook are variants. But do we see the AE ligature is not a variant of A on E, but at the same time is a common visual similarity. I mean, it's not necessary for the AE, but I tell you like in French, the OE ligature on the O followed by the E is a

common variant, similar -- I mean, similar, not a variant. In fact, it's a similar situation. Like, we use that -- you see that for like the word coeur, like heart, you know, in English. And that one because OE was not part of the French spelling for the longest time, and so it got very often transformed into the two letters, which is a mistake, but it's very common. So you have a situation. So again, you have a situation where you're mixing variants on -- confusing on the same level, so you have to be able to deal with that. Next slide, please.

On here, it's very important that we determine level of similarities. So that would help us in the prescreening. We basically, again, we have five level. One is considered identical or near identical. That basically is a [rhyme] of the variant system because basically we're claiming sameness. Two is highly confusable. Again, that is very often also treated as a variant. You saw the example before with the F on the F with hook. And then we get to kind of the gray zone, similar levels on which are most of the time not variants. On very few cases, that could be variants, but most cases are just similar. And then we have the distant where some people will think they're similar, but in most cases, you don't think they're similar. But we still give a weight. And then you have the distinct. This example is kind of showing, you know, on the bottom you have -- first one is highly similar. I mean, it's the same, sorry. That's in Latin and Cyrillic. It's basically a cross-script variant. They use the same letters, mean the same appearance, but complete different meaning. And then you have a Greek versus Latin where the minor difference is a P. It looks like a P. In fact, it's obviously not a P in Greek. And then you have more far-fetched example like P between, you know, Cyrillic because the Y is not really a Y in Cyrillic. And then you have kind

of this funny example like, you know, when you try to do ASCII using [inaudible]. You can recognize more or less it says domain label, but you have to kind of guess it. So nobody would be fooled by that. Next slide.

Yeah, another interesting consideration that there is some simple letters like O, C, L, or I, S, V, X that are common to a lot of scripts, especially the circle. As you can see, the circle has eight scripts that use it. I mean, the one is a bit on this borderline, but the other one really looks like a circle. So if you have a domain that uses just O, you're suddenly confused about a lot of scripts because it's that letter used in many scripts. So you have even some combination that could also exist in multiple scripts. Like we use the COCO example. You can, in fact, create that one in multiple scripts. Although they have no common understanding. They don't mean absolutely the same thing, but visually they look the same. So we have to be aware of this situation. And it gets worse when you have repeated letters like the O, which we do have example, in fact, in the root today. I think we have a triple O. So that triple O is, in fact, blocking many, many, many variants in other scripts. And we have-- so that would have to be considered, obviously, on the fact that Latin has a lot of Cyrillic variants is blocking many Cyrillic today because ASCII is very present in the root zone. Next slide.

So, yeah, this is just a quick example. I don't have to go too much. This is-- on the left is a Latin, ENG. And then on the bottom is a Greek, ETA. On the left is a thing is Armenian character. [inaudible] I think, yeah. And so this, in fact, is showing a bit what your situation perception space where you could have two letters that looks kind of different on the two sides because the leg or the descending leg kind of turn on different way. So you see as more different than the ones in the middle with a

straight leg. So that's kind of showing a bit what would be a perception space issue. Although this is kind of bad example because, in fact, they're all considered variant by the-- in the root zone by the Latin LGR. But anyway, next slide.

So casing, that's, you know, IDN 2008 did remove the case [inaudible] from IDN. And so case [inaudible] is typically done by user agents or browsers or whatever used to access the domain names. And the situation is that we have-- case [inaudible] does introduce some confusability issue, especially for-- between Greek and Latin. Because there are a lot of-- when Greek and Latin are very different in lowercase, they're pretty common in uppercase. The next slide does show that. Next slide, please.

And you see that as an example here. You have the H, Latin H, suddenly move to the, you know, the Greek era and then move to the N in Latin and then move to the, you know, through, again, the chain to a V. So suddenly you have three escalators that are similar to each other through uppercase. So obviously, that would be-- you can't do that. There's no way on earth that we could assume that you can make a blocking situation between three escalators like that. So that just shows the limit also of what you can do if you accept uppercase as a normal case. So we have to be knowledgeable of that, but we also have to accept that we have limit on how much we limit confusability between uppercase because it's just too broad. It creates too many false positives. Next slide.

And then there's a confusable sequence, which is not as explored as the rest. There's less data on this. Obviously, there's a very well-known

example of the RN, I mean, Romeo, November, compared to the M as Mike. And that has been used, in fact, in the past for some phishing on-- nobody knows what Myspace is, but in the old days, Myspace was, in fact, phished by that kind of attack. And then obviously, there's related sequence in-- like I was mentioning before, the ligatures. And then obviously, you have sometimes repeated features. In Myanmar, you have an ornament or a feature that may not be seen because it's kind of hidden in the middle. And then you have this situation where you need a sequence to create a confusable situation. Like in Myanmar, you can put two letters that look like 60, except they're reversed in logical order. Okay, next slide.

There is some also context of script. When you do look at similarity, you have to see them in context of the script. So there's some consideration that's specific to some scripts. Next slide. Yeah, we're always assuming that in a confusability situation, the target user is aware or knows the script. That's very important. And so we always assume that. It's not we're trying to create a global awareness of confusability. But also, we have to be aware that the native reading of script is sometimes also a danger because they do-- their brain basically makes some short circuit on sometimes assuming that things are the same. I'll show that later. Next slide.

Yeah, again, it's again based on who is using it. It's on the eyes of the older, frankly. The good news is that we have a lot of studies done that by some of the GPs. And the Russian panel have done some studies on what could be considered visually similar or not. Next slide.

And so we have determined that we see some scripts are related by history or features. And we see Latin, Greek, Cyrillic. And Armenian, surprisingly, is also related. I mean, they are related, obviously, but it was not necessarily seen as Armenian being confusable. In fact, I'll show some example of that. Obviously, Chinese, Japanese, and Korean. I mean, obviously, Hanja, the Chinese-looking version of Korean, not Hangul, which is very, very different. I mean, there's a few confusion, but very little. And we see all the Neo Brahmi script, the one the script used most in India, which also have a lot of common structure. Not all of them, but you'll see later that there's some example. To throw that, another bonus is the UI, user interface fonts, do create some of their own issue as well, because they try to harmonize and give among scripts that should not be harmonized otherwise. Next slide.

Some correlated scripts, we see Kannada and Telugu are very correlated. There's 34 common shapes between the two scripts. That's pretty high. So more than half. Latin and Cyrillic have 28, which is also extremely high. And that counts the accents, so it's not just the base characters. And then Devanagari and Gurmuki as well. And there's some other case, like Thai allow a very correlated, but in fact, from an origin point of view, but in fact, their shape is quite different, so it's not as bad as it could be. Next slide.

And then we have some weak correlation, but we still have to pay attention to them. We see we have some confusion between Han and Katakana and Hiragana, but those are mostly already captured by variant system in the root zone. Although there's more, in fact, if you go in the visual similar, we may get more of them to be at least studied. And then, yeah, so. Next slide.

Yeah, so the only point down there was the one with this [inaudible]. That's an interesting point, because some scripts, in fact, although they may have inscript, they don't have many, they don't have really cross-script issues on that. We see case for Arabic, Ethiopic to a large degree, and Gujarati, Khmer, Lao, Sinhala, and Thai. Although they may have inscript situations, they don't really have a lot of cross-script issues. Next slide.

Here's some example. You see Kannada, Telugu, we have two strings on. Many people will not see the difference, the two on top. Armenian has a situation that is not necessarily an example. You see the traditional writing Armenian, which doesn't look like any Latin at all. But when it takes a modern form, it suddenly becomes much more closer to Latin. Not necessarily as words. The words in Armenian look very different. That's, by the way, the name of the Armenian, native Armenian. The only letter that is looking like a Latin character, and this one is a U, I mean, looks like a U. But in other cases, you can create basically a lot of Latin words in Armenian that would be, in fact, written in Armenian. On Greek, obviously, depending which script, which style you use, will have more or less similarity with Latin. Like if you take the traditional series, obviously it looks very different. Where if you go on a modern one, obviously more letters will become similar. Ethiopic and Armenian is quite interesting because, like I said before, Ethiopic is not really confused by anything. But if you take one single letter, sometimes you can find confusion. So that again shows an example of when you have very simple two letters or one letter thing, you can get more confusion than you expected.

Okay, so now we're getting to the process part of this discussion. So we saw before scalability doesn't really... we can't do brute force because there's just too many combinations. We're almost infinite. So the only way we can really deal with this is to have a two-step process. The first one is mechanized, and we basically use something based on the variant system. And then we have the second phase, which we call post-screening, where you have a group of expert panels, a panel of experts, that do look at the subset of the labels to do a final determination, if needed. Obviously sometimes you don't need it, but if you need, you have at least the scope of the work is manageable. Okay, yeah, next.

Obviously, yeah, the size. So now, if you slide on the... That's okay, keep going. Yeah, so size of the target also matters. I look numbers here that are probably slightly different from their actual numbers because I use the one where I could read the IDN labels compared to the other one. So I use a source where I could read the IDN labels and see the scripts. So it's about 1500 and some change, entries. Of most of them, 1400 and plus are ASCII entries, and we have 168 IDN entries. So it does show, in fact, the balance between ASCII and IDN today. Interestingly enough, a lot of ASCII entries, in fact, have many variants. They're not [inaudible], but they do exist. Like you can take everything like AAA, ARP, AC, ACO, all those things would be perfectly valid Cyrillic labels.

But the interesting thing, the other way around too works. I found one Cyrillic entry, OPR, that looks exactly like a, we see a Latin sequence. And then we have, again, if we go by scripts, you'd see that most of them are Latin. Two more than the ASCII entry because there's two IDN Latin entries. On CJK again, so 66, mostly Chinese. There's a few Japanese as well. And 36 Arabic. And then it goes down. I mean, there's a few

Cyrillics. Obviously, that means that Japanese. And then you have, you know, kind of goes down to Devanagari, Hangul, and so on. Because India, I think, created one entry for each, for the name of the country in each of their official scripts. So that's why it's basically the whole Neo Brahmi represented here.

So, obviously, from that, you can see that Latin and ASCII are very focused points. We have to be, obviously, that's a vast majority. So the string similarity would be very focused on Latin and ASCII. And the root zone size is small enough that we think that what we're thinking of doing should work. Next slide.

So, again, the plan is to use the variant system to do the prescreening. So the way that in the root zone we do this, we use what we call index label, or variant label sometimes, also called a skeleton in some terminologies. Basically, you associate each label that are variant of each other. You assign a unique value. So you don't have to compare everything to everything. You just assign a value on if two labels are the same index value, they're seen as variant of each other. So it's a very powerful mechanism because it does allow to not have to do a vague computation. I mean, you cannot compare, otherwise you would need more time than there is on the universe to do that. So variant system is very good for that. Yes?

BILL JOURIS: When I read the document preparing for this session, I didn't see anything on how you calculate this index. And I'm having a failure of imagination because I can't figure out how it would be possible to calculate it and get something meaningful.

MICHAEL SUIGNARD: Yeah, we do in such a way, we use the lowest code point typically of any variants. So and then we, and by the way, the index does not have to be a valid label. We don't enforce that. But it's basically the lowest code point that is possible in the possible label. So basically, you take any variant set and you basically always use the lowest code point. It's pretty arbitrary. We could have used the highest code point, but it's a bit easier typically, very often. Yeah, it's easy. Most of the time it's a valid label because of that, because the lowest code point tends to be valid. But it's not necessarily the case. We have some index that could be not valid because it's not a requirement to be valid.

So we do the same thing, because the idea is to do the same thing on the pre-screening, that we basically create clusters and then we basically treat them as a variant set. And we use, and to use basically, in variant we use something that is called a mapping between a variant, a code point on another one, or a code point on the sequence, or vice versa. That's basically a mechanism we use to create the disposition in the root zone. And here we could use that as a mechanism to triage the various labels to see how close they are to each other in the perception space. And so the idea is to use a mix of variant set and use basically mapping to determine which one is a dissimilar, weak, similar, strong, or identical. And obviously if you get down, we give weight to each value and then we do measure the result. Obviously, the bad news on this is that we may include some that should not be considered confusable, because to use that, we have to obviously overshoot the situation. Okay, next slide.

Okay, what kind of data we have? We see we have a lot of data that came from the root zone studies, especially some groups like the Latin GP was very productive, produced a lot of input. We have some input from Unicode, and we have also, but not much about sequences. That's a domain I was not really exploring. Next slide.

And we see the outcome of the string similarity review would be as a, you know, it doesn't pass. I mean, it's not eligible because this was a variant or it was not even valid. On the pass, it's basically, it's not confusable, so it's good. We went through the whole process, through the pre-screening, post-screening, and we didn't find any issues. So it's okay. On the contention is when you have two applied for that are confusable with each other. So a situation where, in fact, it's not, a situation where you have two, you know, two applied for that would be confusable, not with another one already delegated, but among themselves. On fail, we see is when you have a confusion with an existing reserve label. Next slide.

And so, yeah, probably going to pass on those because I don't have much time. Screening heuristic. Yeah, so we use this 1 to 5 again to compare on the, we use, you have to kind of go into the LGR system to see how we do this, but basically we use the same system that we do for root zone to basically compare. And we use both the action part would be used to determine how close they are in the presumption space. Next slide.

We have to, to make that work, we have to do something like we have on the root zone LGR, so we have to create LGRs, in fact, itself. So we would have to take the 26 or so LGR we have on the root zone, and we'll

use a different mapping situation, a different methodology, but using the same structure, the same repertoire, and we'll apply variants to those repertoire and using basically a new set of LGRs that will be internal to these things. They're not going to be produced as a root zone. They will be internal to the mechanism to create those and will basically create a cluster system.

Okay. And so like I said before, we use a cluster index and we collect all existing labels belonging to the same cluster. And then we basically do the analysis of that cluster. Yeah. Okay, that's a lot of text. I'm not trying to summarize that, but yeah, that's another metric. Okay. So this is using another system. We will also consider that. It could be useful in some context. Instead of using the variants system, you would use editing steps. That's based on the [inaudible]. It's explained in more detail in the document, by the way, so I don't have -- It's basically another way to evaluate similarity by using editing steps. For example, you remove ornaments or you even add a letter or you double letters. So it's another system we may think of using to do the similarity prescreening, but I don't have time to explain it.

So we see the post-screening is a manual evaluation. So the idea here, you're going to use a lot of information created by the prescreening to determine, first, that the number of labels you're evaluating is reasonably small, and you would look at the result of the prescreening and validate the prescreening. And then you may even inject into your own label for review. You may, in fact, inject your own. In some cases, people may say, "Oh, you should have also considered that label," because it's not confusable for the prescreening, but we think that, in

fact, it should be part of the pool. So the expert panels, if you want, have some leeway in adding labels to the string under reviews.

The main point here, you go from 1,000 to 1,000, in comparison to maybe a few dozen. So it should be much easier for the panel. And you can see user prescreening principle to do his own study. The panel will not use a focus group, but will use obviously script experts when you have issues with a given script. Okay, next slide.

Sorry, I'm a bit rushing here, but I have to. So we expect to have at least three reviewers on the panel for the post-screening. They need to be familiar with the scripts. They need to have at least some idea what DNS is. They should be able to read the computer score and to override it if they think that the prescreening was not good enough. And the outcome is to reject, add to a contention set, or to accept. And they have to document their choice. That's a key point. You have to do why you took that decision and based on what. Next slide.

I think I'm done. So, obviously, now the next step is to basically affirm what we're doing. And then get public comment on the document and also probably do a trial run on the existing set. It would be interesting to take the 1500 set that we have today and apply similarity review on it to see if we have in the existing set similarity issues. And then we see, make progress. So that's the end. On the next one, just repeat, I think, the links. Okay, sorry.

SARMAD HUSSAIN: Thank you. And we can now open the floor for questions. And Pitinan, if you can help manage the queue. Thank you.

PITINAN KOOARMORNPATANA: Thank you. So please feel free to use the raise hand option if you'd like to ask or comment. So we have the first one in the Q&A pod.

ALAN GREENBERG: Just for the note, on my Zoom, there is no raise hand.

PITINAN KOOARMORNPATANA: Oh, okay. I will monitor in the room physically as well. So first, Bill, please go ahead with your question.

BILL JOURIS: I like the five layers of levels of similarity that you had, I think back about slide 40 or so. But what I'm not sure is how you go about assigning those levels. I mean, when the Latin Generation panel was going through deciding whether things were variants or not, it took us like five years to go through all those. And this would, it seems to me, take a similar level of effort. And I'm not quite sure how that gets done before the next round starts.

MICHAEL SUIGNARD: Yeah. Well, it's a good thing that you'd spend those five years. I know we don't have to spend those five years. So it's our intent to use -- I think the Latin LGR did a lot of work on that. We for sure intend to use the work done by the Latin LGR. We're not going to reinvent the wheel here. So some GPs, the generation panel did a lot of work on this. And we for sure intend to use the output to create our own output. So yeah, it's for

sure. We're not trying to -- there is obviously some arbitrary decisions there. You have to basically sometimes do a judgment call. The good news that, you know, I have been pretty involved on these things too, on [inaudible] as well. So we're reasonably aware. So we see there is some script where we have a bit more knowledge than others, like we see Latin, Greek, Cyrillic. I mean, even, you know, we know pretty well on the -- even for CJK, I pretend to have a pretty good knowledge as well to do -- to look at visual similarity in the CJK universe. Obviously, for other ones, we do expect, based on priority, to ask other experts, especially for the one where we're totally lacking is obviously Neo-Brahmi for India. And also Arabic, I would expect, obviously, the task force, you know, to help us on Arabic because Arabic is a fairly complex subject, and I don't pretend to have any knowledge to do the Arabic part on. We would expect other groups to do that.

PITINAN KOOARMORNPATANA: Thank you. Alan, please go ahead. Thank you. Two questions. The first one, I was somewhat confused on sections 9.1 and 9.2, which are talking about the size of the registry because that's the second level issue. So if you could answer that, I'm a little bit confused. I'll do both questions and turn it over to you. The second one is I was somewhat surprised that given that we're looking at visual similarity and there have been an awful lot of advances in pattern recognition over the last 10 years, I was expecting to see a mention of it, even if it was discarded.

SARMAD HUSSAIN: Just to respond to the second one, I think there's a very clear recommendation to not use an automated tool for this, and we

obviously have to take it through a manual process of review. Thank you.

ALAN GREENBERG: Okay. I understand why we didn't want to use the last one, but the world has changed an awful lot since then. Your phone recognizes your face, which it couldn't do in 2012. Okay.

MICHAEL SUIGNARD: We do mention, by the way, some pattern studies on recognition in words and stuff like that. So there's, in fact, some pointers to that discussion in the guidelines itself. On the size of the registry, I mean, I was just -- I don't know, not sure I got your question right, but we only look at the root zone. We're not looking at second level. This is a root zone list of -- so I'm not sure what you mean by second level.

ALAN GREENBERG: Section 9.1 and 9.2 talk about the relevance of -- for a large registry, this would be complex. And I'm not sure why it's even mentioned.

MICHAEL SUIGNARD: Yeah. Well, because someone is going -- may try to use it, you know. You could try to scale to a larger registry. I don't know if it would work, but at least for now it's not the scope for this application. You're correct.

PITINAN KOOARMORNPATANA: Thank you. Nabil, please go ahead.

NABIL BENAMAR: Thank you. Just one quick question about the Arabic script. As a member of the task force for the Arabic script IDNs, I think that we have done the work that has been implemented in the root zone for the top level domain, and we have also done for the second level for different languages. So I'm just curious on what is still missing in this work so that we can revive the task force. Thank you.

MICHAEL SUIGNARD: Yeah. So I think the scope of this work is slightly different from the earlier scope. Earlier we were looking at variants, which means just the top one from the five categories. Now we are looking at the top three categories at least. So the scope changes. Maybe after analysis the generation panel may or the script community may think nothing else needs to be added or identified for similarity review. But that still maybe requires one pass to just make sure that for similarity review, if there are any additional code points which may need to be identified, they are also included.

PITINAN KOOARMORNPATANA: Okay. Thank you. Bill, please go ahead.

BILL JOURIS: Thank you. I wanted to say first how delighted I am with the document that you folks have put together. As Sarmad and Pitinan are probably more aware than they would like to be, I had a number of issues with

where we were earlier, and the documents address almost all of them. I mean, the capitals and on and on. And I was really glad to see that.

However, I did have one other thing I think you've missed, underlining. Every major browser, Chrome, Firefox, Bing, automatically, when you have a domain name, will change the color, which is not a problem, and underline it. The major word processors, Microsoft Word, Apache, Acrobat, do the same thing. And so symbols that can have a problem with that, for example, there is a -- one of the Latin script members is an O with an underline. If you underline the domain name that's got that, yeah, there are a couple of blank pixels on either side of that underline. But since that character only appears in a few -- pardon me for putting it this way -- obscure languages, a typical user is going to look at -- not even think that he ought to look for something, interruption in that underline. So I think that underline situation needs to get some consideration.

MICHAEL SUIGNARD: Yeah. It's obviously a point we are familiar with. I mean, because we saw when we were doing the root zone. I would probably agree it's probably a sensible point to make, but it will probably be on the context of mostly Latin, Greek, Cyrillic, I mean, the European one, because I don't think it's as much an issue for the -- in general term. But I think it's a valid point. Maybe we should add underline. But that also depends on the mandate. To some degree, we are not really mandated to look too much underline. But if community thinks that we should underline, that's okay with me. I don't have any objection to that.

SARMAD HUSSAIN: Thank you, Bill. That's a good suggestion. And as Michel also said, we are really looking forward to input from the community. So if you think there are certain things we need to add, please do put a public comment, and we will certainly consider them. Thank you.

BILL JOURIS: And I understand it's a work in progress at the moment.

MICHAEL SUIGNARD: Another one also that was mentioned quite often, we kind of pushed a bit, was plural versus singular, because we thought it was a can of worms, because although it looks very promising, it's a very English-centric view of plural. It's not even true in English sometimes, because when they use French words, they use a different way of making plural of them. So plural as a concept is a very dangerous slope. We're not sure we want to go down there. Anyway, so that's something that we may consider, but at this time, we're not considering all this by design.

PITINAN KOOARMORNPATANA: Thank you, Michel. And we are at time, but we will take the last question from Ben. Please go ahead.

BEN MCELWAINE: Hello. This is Ben McElwaine, Google Registry. So it seems like this is basically entirely about visual similarity of scripts. And there's a lot of things that seem to be explicitly out of scope. And I was just wondering if anyone was working on those, because there do seem to be lots of

other potential confusabilities that are maybe even worse, because they would rely just strictly on Latin characters and nothing else. Like, for instance, pluralization. So if you have coin and coins, if somebody's telling you the URL out loud, you might mishear it. And it gets even worse, because a lot of cultures or a lot of languages don't have a concept of pluralization, period. So if they don't speak English as a first language and they hear those, they may not even realize that something is being said plural. So you can imagine they would go to the singular version and hit the wrong thing. So that's one potential issue. Another is just spelling variants by languages, like color, American spelling versus color, British spelling with a U. I think it would be very strange if you had a dot color and dot colour both delegated, just one has a U and the other doesn't. I suppose that's not in scope here, because this is only for visual similarity. But is anyone working on that? Like, are people generally in favor of allowing these kinds of things to be delegated? Or would those be blocked?

SARMAD HUSSAIN: So thank you, Ben, for your question. This is Sarmad. So there is actually a recommendation which considers or is suggesting to look at singular and plural. That is currently under consideration by the community and the board. And beyond that, there are no additional recommendations on any additional scope of similarity, which can, as you are obviously pointing out, can go beyond just the visual part of similarity. You can have phonetic similarity, you can have some other kinds of similarities, grammatical cases like number, but beyond number you can also have gender variation or also case variation. So there are all these linguistic kind of variations in inflectional morphology. But again, as far as this

work is concerned, it's very strictly guided by the scope of policy recommendations which are coming down. That part of work needs to be done at the policy level. Thank you.

MICHAEL SUIGNARD: Yeah, I was going to add also that you mentioned some example of confusability, but there is a whole universe out there. Especially for anyone like me that is reasonably fluent in both French and English, there are so many words that have different spelling, and by doubling their consonants, I get confused all the time because I don't remember which one is the right one. So you have so many examples. You mentioned the O, with OU in color, and all those things. But there are two million examples of confusion using different dimensions. So we have to pick which dimension we are selecting. Obviously we are not going to be perfect, but obviously there is a panel at the end that can also inject, we said, manual input into the process, if you think, for example, you have something that will create such confusion, like color, or something like that. It's always possible to add that in the process in a manual way. So the pre-screening is just trying to simplify, to decrease the size of the study so you can do it in a manual way, because you cannot do the manual study of the whole set. It's just too big.

SARMAD HUSSAIN: Okay, so we are going to close the session. We've run over time. Thank you all for your attendance, your questions, and participation. Thank you, Michel, for presenting. And if you have any follow-up questions, please feel free to reach out to us at IDNprogram@ICANN.org. And please take a look at the study and give us feedback through the public

comment process. We are looking forward to your input. And thank you very much. The session is closed.

[END OF TRANSCRIPTION]