SINGAPORE - IDN Root Zone LGR Workshop
Wednesday, February 11, 2015 – 13:00 to 14:15
ICANN – Singapore, Singapore

UNIDENTIFIED FEMALE: This is the LGR Workshop. We are in Morrison. Start time is 13:00.

SARMAD HUSSAIN: . . . variants and whole level evaluation rules. We will also get some community updates. Igor will be joining us remotely to give us an update on the work being done by Armenian Generation Panel. And we have Dusan here with us who will be giving an update on behalf of the Cyrillic Generation Panel. We also have a guest, Philippe Collin, who will share some application of the data in some other context as well in addition to the root zone LGR. We will then have a question and answer session at the end. So let us start the workshop.

Very briefly, for those of you who may not have the background, basically the LGR project within the IDN TLD program is looking at developing – basically, level generation rules for root zone. These label generation rules for root zone. These label generation rules will allow basically to determine what is a valid top-level domain, and will also determine whether a valid TLD label is going to be – will have variants or not and whether those variants will be allocatable or blocked.

The work starts from developing maximal starting repertoire according to a procedure which was finalized by the community. The Maximal Starting Repertoire is developed by Integration Panel. This work has been largely addressed. And based on this Maximal Starting Repertoire,

community based generation panels work and define the rules for code points and variants and label constraints for various scripts.

We have multiple script-based community generation panels active. Arabic Generation Panel, Armenian Generation Panel, Chinese Generation Panel are formally seated with ICANN and now working towards finalization of their proposals. Arabic Generation Panel has finished the work and completed at least the first draft of their proposal, which they are now reviewing.

We heard updates from Arabic, Chinese, Korean, Japanese generation panels in the morning. We will now also hear updates from Cyrillic and Armenian Generate Panels today in this session. Then there are also some generation panels which are active, which also include Neo-Brahmi and Myanmar; and then Green, Khmer, Latin, Sinhala, and Tibetan panels are also initiating.

Once they complete their proposals, that proposal goes to Integration Panel, and Integration Panel reviews that proposal for eventual integration into what is called the Label Generation Rule Set for the root zone.

To go into more details on the work which is being done Integration Panel and some of the observations which Integration Panel wants to share with the community, let us move to Wil Tan who is the member of Integration Panel and he'll talk about the guidelines for LGR development.

WIL TAN: Hi, my name is Wil. I'm an Integration Panel member. I'm going to just talk through a document that we have published in the hopes that it will help generation panels in their work. We talked about this in a different variation, I think in LA. But this is, in light of the recently published document, which is the Guidelines for Developing LGR, I'm going to just highlight some of the points in there.

The document is called Guidelines for Developing Scripts of Specific LGRs for Integration Root Zone. This is now out for public comment along with two other documents. Let's just delve into it.

Here is just the phases that we envision GPs would go through. Generally in that order, you start with the MSR, then you choose your code point repertoire, find out whether there are any variants and what the variants are and how to dispose of the variants, and then determine if whole label evaluation rules are needed, and then eventually prepare for the submission for your proposal.

Always start with the MSR, which is the Maximal Starting Repertoire. As of currently, we have published version two of the MSR, which includes six new scripts (Armenian, Ethiopic, Khmer, Myanmar, Thaana, and Tibetan). We haven't changed any of the code points [or] repertoires in the scripts that are already included in MSR 1.

When a GP forms, you first select an ISO-15924 script code as your scope. This is what we're going to be working on. That pretty much implicitly restricts the code point repertoire that – your working set would be an intersection of your script code with the MSR to repertoire, with the exception of the inherited code points for some GPs that need it.

The GPs, as part of their mandate, they might be working on some other things. Or you might want to consider a bigger, wider set of code points beyond the MSR, but really for the work of the LGR, we're looking at a subset of the MSR.

Once you have that intersection of the MSR and your script here, the first bullet here with the table is really just an illustration of if you were an Armenian panel, you would say that I'm doing the Armn script code. Therefore, you look in the MSR XML file and look for anything that are tagged with sc:Armn. For Greek, similarly. For Hahn, similar.

Say, for example – some ISO-15924 script codes are actually pseudo codes for multiple scripts. For example, [JPAN], that's actually composed of three different script codes; and therefore you would actually pick any occurrences of any of these three script codes.

UNIDENTIFIED MALE: Igor, you should now be able to hear us. Can you just let me know that you can hear me? Thank you very much, Igor.

WIL TAN: [inaudible crosstalk]

UNIDENTIFIED MALE: Sorry, Igor. We'll get to your presentation momentarily. We'll let the current speaker finish off for the moment. If you could just mute your microphone whilst that's happening, we'll get back to you straightaway.

WIL TAN: [inaudible] that's the outer limit of it. You would then start again—

[crosstalk]

We have interference of sort. We have a translator, perhaps. I'll try to just continue.

Once you have a maximal outer limit, you would then look at each of these code points to see if they qualify for inclusion in your LGR proposal in your LGR. We expect the GPs to positively affirm that each point, if it's – each code point essentially has to be accounted for with some rationale for why they should be included.

If you have a large repertoire like CJK, we don't expect every single one. if you have a block of it because they are part of – as long as you have some sort of a reference that are reasonable, that helps us in reviewing and understanding why they should be included, that's always helpful.

For more info, see the Considerations document. There's a list of references at the end and you can see final documents in the document repository. This is just a very short highlight of the points to consider.

Just a few points to consider when you're developing your repertoire is that many of the GPs probably already are – there are existing IDN tables that are being used on the second level today in many TLDs. Probably that's a good starting point for many GPs. We just wanted to highlight the point that this work is going to – you're going to place your existing table in light of the root zone, which is a shared resource. We value the conservatism principle very much here.

So in view of the fact that the root zone is a shared resource for the entire Internet population, as the [inaudible] states, that actually necessitates a more restrictive LGR so that we can move with caution and everyone can benefit from it.

The root zone LGR, there are also difference between what you put into the root zone versus what you might put into second level IDN tables. For example, the root zone LGR is definitely script-level focused, whereas many of the IDN tables have a very language-based focus.

We frown upon ASCII mixing in the root zone. Also, you may have different variants and [dispositions] in the root zone LGR than what you would use in the second level. So just some of the considerations as you're developing the repertoire and rules.

The next phase would be to determine – once you have that repertoire, you would determine if that repertoire . . . For each code point in your repertoire, you would determine if there any variants, and if there are, how does the presence of those variants affect labels that are being registered? How do they [inaudible] and how do they eventually determine – how do you determine the status of the final labels that are generated out of these variants?

Primarily, we're working with blocked. We're actually just looking at block or allocatable. In fact, the procedure is very clear that we want to minimize the number of allocatable variants. Most of the time, we just want blocked variants. Again, conservatism at play here. So for more info, please see the Variant Rules document.

The LGR process has a mechanism called whole label evaluation rules which are essentially – think of them as regular expressions or patterns that you can apply to restrict the labels that can be generated from your LGR.

You don't necessarily need them in all the generation panels. Some scripts may not need it. If you do need it, then – so it is up to the GP to consider whether WLE rules are required, and if they are required, there are some considerations in terms of balancing, security, and simplicity because WLE rules do incur overhead in terms of your LGR becomes more complex if you have rules in there, because it's not just a straightforward looking at the label, whether that is registerable as a label. You have to kind of process the WLE rules as well.

So it compromises simplicity, but often the WLE rules are there to enhance security, to basically restrict the registerable labels to a smaller subset of the permutations.

In general, we think it's a good thing, as long as it's not overly complex. For more resources, do see the WLE Rules document. It's a bit more technical, benefit the Integration Panel is here to help and happy to discuss any rules that you might thinking of or how you might go about implementing certain rules. We can certainly help with that.

A point to note about coordination between GPs. Michel is going to talk more about that. Essentially, whenever there are scripts that are related, some form of coordination is needed. It's not a requirement of the procedure, per se. We do think that it's a good idea to coordinate beforehand for a number of reasons. So we will encourage all GPs with

[inaudible] repertoires to coordinate with each other early and communicate often.

We envision that most GPs, if coordination is required with other GPs, you would still first work on your own repertoire, your own variants, and your own rules and produce an LGR that is strictly your script.

Then, after coordination, you have this merged LGR that you can present side-by-side. The [lean] one that's just pertaining to your script and the one that's integrated and you can sort of see how integration affects your community. That's what we advocate that GPs should do that.

Finally, we just wanted to explain what we expect to see in the GP proposal. The formal definition is really the XML definition of the LGR that has to be compliant with the machine-readable XML format, which is drafts, [inaudible], IDN tables draft – Internet draft.

In that XML definition, you would have your code point repertoire references. You'd have the rules and variants all in there. Along with that, you would also submit a document basically with pros and texts, explaining your choices and rationale and how you arrived at your decisions for each of the elements that are present in the XML formal definition.

Along with that, we would appreciate – we actually need the examples. If you have variants or if you have WLE rules, then we need test cases, essentially. We need example labels and what are the expected dispositions of those labels so that we can verify – how you think it should behave is how we think it should behave.

Optional – any kind of supporting documents, we appreciate that. For more information, see the Requirements for LGR Proposals document.

Throughout the process, we'd like to know what's the GP thoughts and the process. Keep us in the loop so that we can help when needed. If there are any issues that we think could pose a challenge, we could [feedback] to you early as well.

The procedure is really just [inaudible] for us. Again, always go back to the procedure and read it.

Here's a list of resources. All the documents are there. You can get to it by going to the Root Zone LGR project Wiki, and then going to the document repository and all these documents will be there for you to download.

That's it. Thank you.

SARMAD HUSSAIN: Thank you, Wil. We'll move on. Michel will be talking about how to design variants and WLE rules. Michel?

MICHEL SUIGNARD: Good afternoon. To save some time, we had some overlap between Wil's presentation and mine, so I'll try not to go over the same points. I'll probably go a bit faster in some slides.

We are going to talk first about the variants, some basics on them. First, we will see what variants are. Perceived as being the same character by user community. So it could be the same [inaudible] a lot of different

ways. It would be [inaudible] identical, but very visually different. A good example for that would be traditional versus simplified Chinese, whereas they look different, but the user community perceives them as being the same thing.

Other case, you have similarities from an appearance point of view. You would be basically looking at two characters. Your cross-script, for example, that looks exactly the same, but in fact, it belongs to different scripts because the root is a shared resource, again. Even if it is from different scripts, we may have to take that into account when we do integrations.

A good example of that we see is Greek, Cyrillic, and Latin, which is a people example of seeing a lot of variants that do go across scripts.

Important point that many [inaudible] LGR don't have to worry about variants is not – everyone does not have to worry about them. We see many, like Arabic LGR has variant issues. Probably many in CJK would have the same thing, but we know many of the scripts where it will not be an issue whatsoever. It makes their life quite a bit easier.

Again, variants [live] [inaudible] in the root zone. So we have no idea of languages on the script. Is it script-based? Even in some cases, we're sharing the script among multiple LGRs. So we have to take all those things into account when both the Integration Panel is looking at them [on] where you're doing your own work on defining those variants.

Despite apparent restrictions, due to these blocked variants, you have to understand that you define the variants at the code point level, but when you do the disposition of them, we do that at the label level, so

it's very different. You're only blocking a label that has exactly the same content, except for those variant code points. It's not as bad as it looks at first approach. You still have thousands of [variable] labels, even for an existing writing system. You just have to add one single character, then it is a different label. So you're not re-blocked by this variant disposition.

Some requirements for variants. They have to be symmetric. So if you define variation A to B, you have to also B to A [inaudible] on the fact that you have more than two, all the relationships have to be defined. It's very important [inaudible] Integration Panel, we will verify on [inaudible] those rules. There is no "if" – those have to be respected. Your variant set is large for given [inaudible] number of correlation can be, in fact, [printed] out.

Variants that intersect scripts must be defined on each of those scripts. That shows, for example, when you have a shared homoglyph among scripts, that means that we cannot do the work on one of the scripts I the work is not done on all of them. That's a good example between Latin, Greek, and Cyrillic. Those have to be worked in parallel. We cannot, in fact, integrate one of them without the other two being done. It's a real clear requirement on that one.

You can see I put an "O" there. The "O" obviously looks exactly identical between those two scripts, so we would need a variant disposition for all those three scripts.

So, some description of the types. Variants can be in repertoire within a single script. It does happen. We have that situation in Arabic. We are seeing, in fact, quite a few of those where you have characters that are

very similar in appearance, almost identical, especially in Arabic because you have positional forms. Things that are a bit more complicated, because you may see on the [inaudible] that they look very different, in fact. But if you take the [medial] form, they may be in fact exactly identical.

Same thing for Chinese. You may have traditional or simplified, in fact, part of the same repertoire. They may be out-of-repertoire or across scripts. Out-of-repertoire, I mean, for example, that you may have variant relationships that do affect some of the Japanese characters, but in fact the variant set is in fact also includes characters they use in China. Those kinds of situations would have to be taken in context by both the Chinese [inaudible] Japanese, even if one character is [inaudible] repertoire.

We see across script. That's the case I already described before, which is Greek, Cyrillic, and Latin situation.

Types assigned to variants drive disposition for labels containing these variants. We defined for each code point that have [inaudible] variant set, you will define the type of variant.

Typically, we use blocked or allocatable, but in fact, you can make that more complex. If you look, in fact, at the XML LGR specs, there's an example there of using traditional and simplified where to minimize a number of allocatable variants, they do one thing, which is to limit the set to – is it all simplified, all traditional, or the original label that was applied? So [inaudible] different types of that.

[At the end], when you get to the disposition, you see that [there are] only two ways to dispose of it: blocked or allocatable. They can also be invalid. That's another way. Because obviously if you have an out-of-repertoire variant in the label, when you try to submit, that's going to be considered as invalid because it's out-of-repertoire.

On the use of allocatable variants, as we know, [inaudible] procedure says that allocatable variants are basically to be – may be allocated to the applicant. I'm repeating the terms in the procedure. So there's a very clear intent that the allocatable variants are in fact to be allocated – [inaudible] location of delegation is totally used to of scope for us. We have nothing to do with it. We are just busy creating the possibility for those labels.

[inaudible] allocatable is that you're in repertoire. That is a given. But variants are inherently are the 'same' character. They're not just homoglyphs. They're also the same from all purposes to the user.

On the third point, which is kind of sometimes a bit more difficult to determine, but it's an important point nevertheless is that allcoatable variants – we need better [use] for cases where it's difficult to enter the alternative, where you only have one choice among many of the variants that could be existing for one character.

We have some examples again in the Arabic situation where you have the Arabic [inaudible] where it's very difficult to get either one in your own keyboard. I don't want to go too much in detail.

On those input mechanisms or input limitation, [inaudible] key aspect of why you may want to do allocatable.

Some cases, in fact, may be treated by not using variant [inaudible]. You may even think that maybe if there is a slight difference, maybe you kind of want to leave the [confusability] issue to a higher level, to [let] the string review go with it or not try to [inaudible] the rules in the lower level.

Some situations are complicating. It may not be possible to establish simple rules or because [inaudible] create too many [variants].

On the last point that we see, [we only say] they may be allocated in the procedure. We don't say anything what's going to happen beyond on that. As many of us know, creating a allocatable variants is really going to be complicated. DNS is not really designed to do that, especially if you start [inaudible] under the root with variants that look the same but have been [inaudible] maintain the same [inaudible] for all the different levels of the sub-zone of that root entry. We want to be conservative of the number of allocatable variants you create.

So, some examples. I used Greek here to give you some idea what that means. In repertoire, for example, will be a Sigma 'σ' versus final sigma 'ς', which would typically been seen as probably variants of each other.

Variants with Latin, those would be out of repertoire. On variants with Cyrillic, those would be again out of repertoire.

I do provide [inaudible] to the Greek LGR. It's just a sample, because we are not in the business of creating LGRs, but we are making them as examples on [inaudible] relationship.

Another one that is different is variants for integration [inaudible] Japanese case.

Some of what I am saying on these slides may not correct because we had a discussion with the Japanese GP in the past days, but we still don't expect the Japanese LGR to have many variants, if they do. At least at this point, they may have some, but very little.

The situation, though, is when they do the integration, they are going to get a variant set by basically absolving some of the Chinese set. That is something that we'd have to coordinate and get a common view with the Chinese set.

The variant set for the common repertoire to be the same. On any character [inaudible] variant set which is [partially] in their own repertoire, they have to add the other repertoire into their own LGR.

It's quite a bit complicated for the non-CJK folks around here, but we have been through some of the details with the CJK panel. I think they understand what we mean here.

So the end result may result in some situation where [inaudible] will still nevertheless be blocked in their own disposition.

Important point. Each LGR is still has total control on the type of variants and the disposition of them. So they can define totally different types or they can define totally different dispositions. There is no need or requirement to have common disposition [on types]. The only requirement is to have the same content for the variant set for every characters that do intersect between the repertoire.

I gave an example here. Down there, there are four characters. They have some variant relationship in China, so that means that Japan also will have to do something about them. Those four characters are likely

to be included also in the Japanese LGR because they're part of the basic set for Japan.

This kind of describes the process how to do the work here. It's a very important point. The first one says you [really] do your work on your own first. You don't have to coordinate for the sake of coordinating. First, you have to do your own work, which is to define your own repertoire independently of other groups. You determine your own repertoire, you determine your own variant step. That's your linguistic [correct] view of your LGR.

When you're done with that, then you go [over] the groups and you compare your repertoire and you compare your variant sets, and then you do basically your merge.

This is not necessarily a simple process. We, Integration Panel, have some expertise to help you on those aspects.

I'm asked to speed up, so I have to go a bit faster. I'm done mostly with variants. We're going to jump on WLE rules.

The first point was already mentioned by Wil, so I'm not going to repeat that. Second point was why we do use the [review] [inaudible] whole level evaluation rules is mostly to determine required or prohibited context, so things that don't make sense in a given writing system. We are not talking spelling here. We are talking more like things that don't make sense, [inherently] the writing system. Also, restricting combining sequences in alphabets and enforcing simple composition rules [inaudible] abugida.

A quick example is combining Macron Below. Most of the time, it's used only in pre-composed characters, but there are some languages, like in Africa, that needs a combining sequence. So you can, in fact, create a rule that basically only enforce that the combining sequence can only exist for some contacts. In this case example, we use [for them] that this character, this sequence with Macron Below can only exist when it follows 'c', 'q', 's', and 'x'. And you can [inaudible]. It's pretty simple, in fact.

Another example is Thaana. Thaana is, in fact, a script that is written in syllables, but encoded as an alphabet. It is kind of [inaudible].

So we can, in fact, enforce [inaudible] very simple set of rules that enforce that script formation by basically making sure that a consonant is always followed by a vowel and only one. [inaudible] following each consonant.

Conclusion. Very complex features, no doubt about that. But as we go, we're getting more and more experience on dealing with them as a group, as the Integration Panel has been really facing a lot of these issues with very different scripts. We can really help you. It's very important that you work with us to understand how to do things to get a better chance to be accepted.

So there is, again, under resources. Wil mentioned the [inaudible] document variants, WLE rules. Those go in some details on some of those examples. I have also put as an example the Thaana and the Greek LGR. Those are full-blown LGR examples. They do contain what you need to do in LGR. They're simple, but they do contain in fact the Thaana WLE for Thaana with the full-set syntax for it.

On the Greek, same thing. We added some examples of how you do out-of-repertoire variants, on how you implement that. So you can use them as examples how to do things as a tutorial to create your own set.

Okay, that was fast enough. [inaudible] open for questions. Okay, thank you.

SARMAD HUSSAIN: Thank you, Michel. The next presentation is by Igor. He's going to be joining us remotely and he's going to be giving us an update on the work being undertaken by Armenian Generation Panel. Do we have him online?

IGOR MKRTUMYAN: I am here. I am online. [inaudible].

SARMAD HUSSAIN: Okay. Welcome, Igor. The floor is yours.

IGOR MKRTUMYAN: Hello to everybody. [inaudible, recorded too low]

So the Armenian language is a European language spoken by Armenians, so it's an official language in Armenia. [inaudible] spoken throughout the Armenian [inaudible], and today it's widely spoken in the Armenian [inaudible]. Next slide, please.

So Armenians have Armenians has its own unique script. The Armenian alphabet, which was invented in 405 AD by Mesrop Mashtots. So it's a

kind of independent brand of Indo-European language family. There are two standardized modern literary forms, Eastern Armenian and Western Armenian. [inaudible]. Next slide, please.

Here we see the [inaudible, recorded too low]. Next slide, please.

SARMAD HUSSAIN: Go ahead, Igor. We are on the next slide.

IGOR MKRTUMYAN: Next slide, please. So we started [inaudible] problem with visual similarities. So there are some visual similarities between Latin, Cyrillic, and Greek alphabets. Next slide, please.

This is our generation panel. I am the chair and we have seven experts. Next slide, please

We started with [inaudible] of MSR-2 for Armenian script. The next step was analysis of visual similarities in Armenian and scripts having commonality with Armenian.

We started development of presentation for this workshop, and then we are planning to collect community opinions and remarks, and development of a final report to Integration Panel. At the same time, we will come to a final decision on LGRs for the Armenian script. Next slide, please.

This is the schedule of meetings. We are already on the fourth meeting. We'll continue and we hope to finish and submit the final report to the Integration Panel on March 31. Next slide, please.

Armenian Generation Panel mailing list was created and here is some information about that. Next slide, please.

This is the Armenian MSR-2 table. We accepted the document published in MSR-2. There are [38] scripts in Armenian MSR-2 table. Next slide, please.

Here are the results of our visual similarity evaluation. The first table is between Armenian and Latin. We see the Armenian scripts and the Latin scripts with visual similarities. The problem exists and we continued with starting visual similarities with the other scripts. Next slide, please.

So here is a table of visual similarity evaluation between Armenian and Greek. There are some similarities. Next slide, please.

And here is similarities between Armenian and Cyrillic. Not much of them, but still there exist some. Next slide, please.

And here are similarities within Armenian. There are some strings that will be a bit difficult to differentiate which scripts. Here on the left column there are examples of such strings and the corresponding scripts.

There are also similarities in scripts in Armenian. As you see, there are several scripts that are similar to other scripts. Next slide, please.

So here are the conclusions that we've seen. There are two standardized mutually intelligible modern literary forms, Eastern Armenian and Western Armenian, with different orthographies. So,

semantically the same word can have different spellings in Eastern and Western Armenian.

But we came to opinion that will not conclude to any LGR. As a result, the Armenian Generation Panel will not address in the LGR document issues arising from the different orthographies. Next slide, please.

Until now, we decided that Visual similarities will not be reflected in the LGR for the root zone. They will be rather be solved by mechanisms beyond the application of the LGR. The problem will be solved by limiting Armenian domain names strictly to the Armenian MSR table, Latin dash and Latin numbers. Next slide, please.

We anticipate that the relationships with the related scripts would

not affect the content of the Armenian LGR. Visual similarities of related scripts will be blocked by the domain registration program as it will check the scripts for the correspondence to the Armenian MSR table and will not allow domains names with visually similar code points of related scripts. We are not sure that the same blocking mechanism will be implemented in other IDN domain registration procedures, but we can recommend that to the corresponding IDNs. Next slide, please.

[inaudible] an understanding the visual similarity of strings and scripts within Armenian IDN can be used by domain registrants for phishing or registering a domain similar to a brand domain. We cannot set any rule forbidding the visual similarity of domain names as there is no way to distinguish whether it is normal or intentional because [inaudible]. There can be thousands and thousands of brand names, trademark, and

company names that we cannot analyze in order to find out whether it will be used for phishing or some intentional [malintentions].

So that's all for today. We hope for the community participation and we are willing to receive Generation Panels feedback, their opinion, and their advice and remarks in order to go forward and improve our report to the Integration Panel. Thank you very much. That's all.

SARMAD HUSSAIN: Thank you, Igor. We are running a bit late, so we'll move right to the next update from the Cyrillic Script Generation Panel. Dusan is here with us and he'll take us through the presentation.

DUSAN STOJICEVIC: Hello, everybody. I'm sorry Yuriy isn't here. He got the flu, so I will be replacing him.

Cyrillic Generation Panel was created during the last meeting in LA. The result of the work of this panel will be the proposition to MSR-2 and propositions for Cyrillic script code point repertoire for LGR. Script codes the panel explores is #220 Cyrillic according to ISO 15924.

As you can see, in total, there is 13 countries and 108 languages that form geographical territory from [inaudible] in Europe up to Mongolia and West Pacific Coast with the use of Cyrillic scripts.

The panel took in consideration the following languages: Belarusian, Bulgarian, Macedonian, Mongolian, Montenegrin, Kazakh, [inaudible], Russian, Serbian, Tajik, Turkmen, Ukrainian, Uzbek.

To simplify the work, we divide these languages into four territories or areas: Balkan, Russia, Ukraine, Belarusian. The third one is Middle Asia and fourth one is Mongolia.

On this stage of work, the research is of 95 ethnic minority languages in general in Russia [were] conducted. The panel proceeded from the proposition that Cyrillic structurally and historically related with Latin and Greek, but more details should examine during work of the panel.

This is the general structure and you can see the close date. The close date will be 16th of March, and [staff reports] due the 6th of April. I am trying to speed up a little bit.

This is the table of confusion variants in Greek and Latin point code tables for searching across script confusion variants. The Latin script presented the main option and expanded option for IDN.

This is the [inaudible] result of work of Balkan, Russia, Ukraine, Belarusian small groups. The Greek scripts were analyzed [case] for presence of cross-script homoglyphs.

Same table for Latin script.

And this is Yuriy's favorite. A few words on homoglyphs and punctuation. So in Ukrainian and a Belarusian languages apostrophe is not a sign. Apostrophe is a letter, so it cannot be classified as a type of punctuation. You can see how this sign is used in Ukrainian and Belarusian language.

For the conclusion, there is six bullets. First bullet, A) considered confusion options only for cases of cases of "external" cross-scripts.

B) has done work which gave preliminarily results for cases of confusion variants in two regions within (Balkan, Russia, Ukrainian/ Belarusian languages).

C) we cannot form a complete and balanced position to do full public comment version 2 of the Maximal Starting Repertoire (MSR-2) at this moment, but the Cyrillic Generation Panel will make all possible efforts to make its proposals in time.

D) we do not have data on the analysis of possible options for the confusion variants for two regions within Cyrillic scripts: Mongolian and Middle Asia

E) however, the unit will prepare some recommendations based on available data

F) potentially can to form position on develop policy recommendations which can form base for LGR.

That's all.

SARMAD HUSSAIN: Thank you, Dusan. Let's move on to our next presenter, Philippe Collin. He's going to be presenting on behalf his organization. This particular work is not related to LGR or the IDN program work which is being done at ICANN. It's independent work. They're here to present what they're doing and share it with the community.

PHILIPPE COLLIN: Thank you, Sarmad. My name is Philippe Collin. I'm here to represent OP3FT. OP3FT is the promotor of a technology called Frogans technology, and that is a new class of Internet sites. I'm telling you for the context, I'm here to replace my colleague. Ben Phister was a participant here at ICANN in Los Angeles and I will be happy to answer your questions after this presentation.

First of all, the Frogans sites themselves, they're a class of Internet sites that display and navigate the same way on all devices with the help of a little application called the Frogans Player.

The beauty of it is doesn't mean to replace the Web in itself, but to create a new class of sites. I'm just going a little faster here.

The way we address each site is through a Frogans address, and that is basically composed of two parts. The Frogans network on the left – actually, before the star sign, which [inaudible]; and the site name which is after the star sign, the separator.

So you see here two types of programs addresses, Frogans* which is the public Frogans network; and on the right, network-name* where you could replace network-name with any name, either trademark if you're a trademark holder or generic name or name you invent.

The interest and the reason why I'm here today is that all those addresses can be written in a variety of writing systems from around the world. So we're not saying it's only Latin, but it's a combination of those.

Here are the current ten linguistic categories that we do accept as Frogans. I see that we here on the panel have the same

[preoccupations] when it comes to confusion. But Latin of course, Chinese, Cyrillic, Japanese, Korean, Devanagari, Thai, Greek; and on the right side, Arabic and Hebrew.

So basically, you can compose a Frogans address with characters from all those linguistic categories and we make sure – we try to make sure – that they're not confusable for obvious security reasons.

The way we do that is through a selection of employable characters and also a set of rules, and those rules can be of many kinds. Here we have a few examples. One that Michel mentioned earlier, the difference between traditional and simplified Chinese, so we do sort that out. One that also was mentioned earlier, the use of letters that are the same in two different [inaudible]. For example, here the Latin A and the Cyrillic A. But also inside a given category where we try to sort out the confusion. For example, here in Latin between the big "i", the 1 number, and the lowercase "L."

Basically, think about a big unique registry with addresses in a variety of written scripts. We had to come up with a way to avoid confusion and we developed a model to deal with visual and semantic confusion, and it's actually a two-part model. Here it's a little more specific to our solution, but it will help you understand.

The first part is called IFAP, and that is to make sure that the address respects the pattern. Basically, there's a network name before the star and a site name after the star, whereas FACR is the set of rules that are related to languages themselves. So maybe in this group here, you'll be more interested in FACR.

Actually, we are also interested in what you have to say for each of the rules in FACR. For example, I'd be happy to talk to people who work in the Cyrillic or Armenian to understand how their rules are compared to ours.

The whole idea here is to make it easy to recognize an address and mas that that address is unique and cannot be confused with another one.

Of course, because of the number of linguistic categories, we have a number of sets, what's called valid network names. So Latin [inaudible], in some cases, they do overlap, so we had to come up with rules that help understand which category, which linguistic category, a given address [whether it] be the site name or the network come from.

Just to make it more clear, I'll present a few examples. I hope you can see here on the screen. Let's go with the first one. And of course, those examples might be well-known to you and we'll see how it progresses, if it rings a bell.

For example, the first one, we use the letter "e" with an accent, which is a letter in Unicode, but we don't accept the combination of "e" and the accent. That's in Latin. When we see those, we say the network name is not compliant in the second column here.

Second line, when it comes to confusion between the "hello" on the left in Latin and "hello" using the Latin letter [iota], then here we reject the second form because the Latin letter [iota] is not an employable character.

Third case is [inaudible] in Latin "PayPal" and PayPal using one letter borrowed from a different linguistic category. In this case, the "a" from

Cyrillic. In that case, we say it's not [inaudible] because it's not valid. We cannot do that. We cannot use a letter from a different case.

Also, there's another rule where [inaudible] switch. Go a little further. Oh yes, [strasse]. Here we consider the sharp "s" in German as being the same thing as being two s's, of course as far as name resolution so that you cannot register those two names independently. They are considered to be identical in our view.

There's another example here with "hello". That's a well-known example where the capital "O" on the left and the "0" on the right. Those are considered to be confusing because they are similar visually. In this case, what we say is they are convergent. So we do calculate a technical form called the conversion form for each of those network names, and if the technical form happened to be identical in both cases, then those two names are deemed to be convergent, and therefore cannot be registered separately.

There was a lot of Latin here. There's one more here. On the left, you have "amis" which is visually confusable with "arnis". So in this case, we say "m" is equivalent or convergent with "rn" and therefore those two forms cannot be registered separately.

In a different category – for example, Chinese – I'm not a Chinese speaker myself, but it looks like some characters look exactly the same, depending on different scripts in Japanese. In this case, because the two characters, the 3708 and the 37D8 look the same. We consider they are convergent, and therefore cannot be registered separately.

The next example was mentioned by Michel earlier, the use of simplified Chinese and traditional Chinese. Here, the two characters are not visually confusable, but because they mean the same thing, they're semantically identical, then we consider that those two forms are convergent.

To finish, just two more examples. Here we have what we call the inter-linguistic category convergence which aims at identifying two forms that are completely written and using scripts from different linguistic categories but still could be confused. The two examples are scope in Latin, and the equivalent in Cyrillic which looks the same but is formed by five letters that are not Latin. In this case, we calculate the inter-linguistic category convergence form and we determined that they are identical, and therefore cannot be registered separately.

Remember, we have one unique registry, so we have to make sure that every single form is different from the next.

Last example. Here we combined two types of rules. If you take BEAT in Latin, that can be written in Greek with the same visually identical letters (BEAT). Of course they're different code points. But because BEAT in Greek can also be written in lower-case – and I will call that beat as well, even though they're not Greek letters – those three forms are convergent and therefore cannot be registered separately, which means that if you look at the one on the left and the one on the bottom, they look different, they're used in different terms, but still they are considered as convergent because they're both convergent with a common form.

Thank you. Just to reiterate, we are interested in working with specialists in each linguistic categories, so if you want to see me after this presentation I'll be happy to talk. In the slides, you have resources online to know more about the project and know more about how we could work together in making those rules better and more useful. Thank you so much.

SARMAD HUSSAIN: Thank you. We actually exhausted our time allocated, but we do have the room for another 15-20 minutes at least. So if you have any questions to the panelists, I think we can. Please use a mic and let's address at least a few questions before we conclude.

JANICE DOUMA LANGE: Sarmad, if I could just say while you're waiting for the first question, the remote participants, of which there are 13, wanted to thank each of the speakers. Along the way, they each wanted to thank you all for the information. I thought I'd let you know. Thank you.

SARMAD HUSSAIN: Thank you.

MEIKAL MUMIN: Hello. Yes, my name is Meikal Mumin. I'm part of the Arabic Script Generation Panel and TF-AIDN, and I had a question for the representative of the Cyrillic Script Generation Panel.

You mentioned that you have a number of languages which you are looking at which you're studying and considering the repertoire for integration, but you also mentioned some 95 ethnic minority languages on which no research was conducted. So I was just wondering if you were publishing some information on which languages you looked at, which languages you didn't look at, and qualification and why you did that and why you did not do that eventually. I think that would be helpful. Thank you.

UNIDENTIFIED MALE: Thanks for your question. We looked at the Russian language. In Russia, you have this 95 minority languages with Cyrillic scripts. So we looked at general Russian language and general Russian script. This was the basics of our work. But we will look at every minority language under Russian languages.

UNIDENTIFIED MALE: This is [inaudible]. I'm from Pakistan. I'm working for Arabic IDN. My question is, as a community, we are looking to accommodate more and more languages. The Integration Panel is looking for the security. So what would be the [inaudible] situation where we can accommodate more and more languages not compromising the security, without compromising the security?

UNIDENTIFIED MALE: There is obviously a balance here to establish between scope or breadth, if you want, on security comprehensive. The thing is we're [on]

the root here. We're not trying to define LGRs for second zone. So you have way more capability, if you want, to represent more in subzones.

The root is really not going to be used – you're not going to have hundreds of thousands of entries in the root. It's not going to happen. There's [a cost] to create an entry.

It's also important to maintain security. The more sometimes [inaudible] languages introduce a security issue through the confusable. There's only so many ways to represent, especially on the Latin set or even Cyrillic. You have a lot of characters that look very similar. On some of them, a minor variation. So the more you introduce some of those variations, the more you introduce security issues.

I don't have a black-and-white answer to you. You just have to balance the need to be comprehensive on the need to be secure. We try to restrict to – in the way we described the MSR, in fact, that was one of those elements we used. We were trying to have some notion of usage. It's not just because a language exists. It also have to be an everyday use. We're not trying to be a collection of academic correction everything that was ever done on Earth. It has to be used today.

We're not trying to – the root is not really a linguistic repository. It's there to be creating [inaudible] that can be used in a safe way. There's some balancing constrain on what we can do. I think we [inaudible] that or so on creating the MSR. To some degree, [inaudible] the MSR is some sort of sandbox. You have to stay within the boundary. We cannot help you by [inaudible] taking those considerations to account.

So as long as [inaudible] is on the MSR, you have a good starting point to do your work on. But you still have to justify why you want that. Again, sorry, I don't have really a good answer for you, but it's really basically a judgment call that all of us have to do at some point, how far you want to go in breadth [inaudible] compared to [inaudible] security issue.

UNIDENTIFIED FEMALE: We actually have a question and an answer on the remote participation here. The question is from [Yoshito] [inaudible]: "I need concrete reasons for not mixing the ASCII."

SARMAD HUSSAIN: [inaudible] answer that?

UNIDENTIFIED FEMALE: Right. The answer is actually posted on here as well. From Asmus: "The procedure [inaudible] established that the IP gets to make the decision. In case the script mixing needs to be allowed in some additional cases, but none were foreseen."

SARMAD HUSSAIN: Michel, do you want to comment on that further?

MICHEL SUIGNARD: Yeah. For us, with ASCII mixing, there's a reason [inaudible] add ASCII mixing, you're putting your own script in the mix of the Latin, Greek, Cyrillic confusion study. So you're basically putting a big monkey wrench

in your own LGR formation, because now you have to coordinate with three additional LGRs. That's kind of a process issue, obviously.

But at the same time, we have to at this point draw a line. I don't really think that most LGRs need ASII. They can do that on their own ccTLD or in subzones. But the root is a shared resource.

It's the same way I saw in a document where I was seeing a digit being proposed or a dash. This is not possible. We don't allow obviously – don't even come to us with a proposal to have digits or dash in the roots, because this is a [inaudible]. MSR is your sandbox zone. Please don't go outside of it. We don't have dash or digits in the MSR.

SARMAD HUSSAIN: Thank you. Edmon?

EDMON CHUNG: I guess I'd like to offer an observation in terms of the Integration Panel and your work on this. I think the MSR-2 is great and you've spent a lot of time around it. The Bible, as you referred to it, are the procedures that were created through the community over a long time. It's a very solid document.

However, I guess it's probably not very useful or sometimes problematic to refer to it as a Bible, both in the sense of it and the name of it, because part of the program – I understand that within the current timeframe and all of that, we should play within this structure and procedures. However, part of the program is to review the entire process as well in a little while.

I don't think it is conducive to the dialog if we keep being overly prescriptive on certain things. I understand why the motivation and all that, and I support those motivations. However, if you look at some of the generation panels versus the others, especially for those who haven't been in the discussion as more and are just coming in, they need space to explore the linguistic issues on their own a little bit before being overly prescriptive on it.

Certain scripts – for example, Chinese, I take it as an example – being prescriptive would be very useful for those cases. I sometimes talk to the Koreans, and when they consider Hanja, it might not be as useful being overly prescriptive because they haven't quite explored the entire linguistic issues yet.

We probably come back to those parameters and those issues, and when it comes down to actually making decisions for the LGRs that are published, those will certainly be very useful and very critical. However, to allow for the community to explore what is needed from their own language generation panel, I think that's actually more conducive to the overall conversation. That's my observation.

Then I actually have a question. It's interesting, the Armenian panel raised a rather interesting question. They looked at the Armenian script versus Latin, Greek, and Cyrillic. I know the current thinking is that the Latin, Greek, Cyrillic needs to be kind of lumped together. It kind of brings to question how does the Integration Panel decide which scripts – the Han script might be easier because it's kind of unified. The others might be more difficult. Is Armenian supposed to do that? What about Armenian Georgian? What about . . .

So I was just curious how the IP would try to make those determinations. Thank you.

MICHEL SUIGNARD: To some degree, that's why Integration Panel we have encoding experts. I've been working in Unicode [inaudible] 20 years, so I guess I have some knowledge of what it takes to do those kind of determination.

In the case of Armenian, in fact it's reasonably simple because unlike Latin, Greek, and Cyrillic when you find in fact in a given device those languages, you have total perfect homoglyphs between those three and you can make full words. It is very easy to make a full word in [inaudible] languages that will look exactly the same. Piece of cake. There's hundreds of them you can make.

Between Armenian and [those], there is basically zero. There is almost no opportunity. If you want the intersection between Armenian on any of those [inaudible], it's so limited. In fact, you even need a special font, because most of the fonts don't even [inaudible] not making the same. They were similar, but not the same. With few exceptions, [inaudible] "O". The "O", there are not that many ways to do a circle that means "O", but everything else you have something think about changes.

If you have a full word, you're not going to [inaudible] a lot of confusability. If you have [inaudible] confusable, they're going to be [cached] basically at a higher level, in my opinion. They're going to be [cached] by the [inaudible] panels that you have in ICANN when an application is being [judged]. It's something that's really in your face.

But Armenian does not really have confusability at the kind of level with those scripts that Latin, Greek and Cyrillic have. We see it as a judgment call. I recognize this is a judgment call. We have to exercise it at some point.

But in our opinion, Armenian does not have to do confusability coordination with Latin, Greek, and Cyrillic. It's not really justified. It's basically a judgment call that within the IP [we are] making.

To answer your first concern, to some degree, to make this whole process work, we need to have a framework. So for now, we have a framework. We are using it. If the community wants to revise it at some point, we will accept it. But we have at this point to work within the framework which is basically establishing an MSR within the procedure when we have an issue. We go through it and we basically use it as a way to answer concerns.

As long as that is [a law], if you want, we have to follow it. If the community change the [law] on us, we follow the new [law]. But we need a framework. We can't just work with totally open question on basically – I think document is very helpful for us to be able to make progress.

EDMON CHUNG: Just quick. Yes, I appreciate that very much. Obviously, personally I fought long and hard for that procedure to be in place. I'm not about to change it at this point.

However, I similar questions or similar requests that violates the procedures keep coming up, yes we should say that this version doesn't

work, but we should also take those issues down, and when it comes to review, maybe taking a look at whether changes are called for. That's [the point].

SARMAD HUSSAIN: Yes, thank you.

UNIDENTIFIED FEMALE: Okay. We have—

SARMAD HUSSAIN: Excuse me. Dusan wants to comment on that.

UNIDENTIFIED FEMALE: Of course.

DUSAN STOJICEVIC: I will add from my presentation. The point that we start with Latin and Greek is historical. This is the point why we chose this. But this is no [inaudible]. We cannot finish only on that variants, and this is not the full job.

SARMAD HUSSAIN: We'll take one more question on remote and then we'll conclude.

UNIDENTIFIED FEMALE: Thank you. Question from Peter Green: "For [script visual] similarity, does each script panel have uniform standards to determine script

visual [similarity]? For example, using tools and metrics. Or does each panel have its own way to analyze visual similarity? Thank you."

UNIDENTIFIED MALE: First of all, the first responsibility is [inaudible] generation panels. We are just kind of busy looking at what the proposed LGR. We may have our own opinion, but [inaudible] generation panels need to have their own opinion on what is similar. They have to do basically their own work on variants. They have to basically make a list. They have to negotiate them with those [coordination] scripts, like Latin, Greek, Cyrillic. They have to agree basically on a common set.

Then they submit their work. They may [inaudible] ask our opinion, but our opinion is just that. It's just our opinion. But at the end, we're going to verify the end result. There is obviously some judgment call on what is similar or not, especially between Greek, Cyrillic, and Latin. The similarities, sometimes it's in your face. It's 100%, [inaudible] very common [fonts] that are widely available and you find 100% similarity.

For some of them, it's not that obvious. You have 80% or less. There's a judgment to be made there. Again, you don't have to make variants everything. There is other processes beyond the LGR to do that. If you have confusability, at some point, this is going to be addressed by a protocol that is above the LGR formation. [inaudible] domain is proposed for allocation or delegation, there is a lot more processes going to be applied. [Some of them] may object for name confusability issues.

So we don't have to capture everything. We have to capture the obvious cases that really should be done mechanically, so nobody has to spend time or energy to study for those. But there will still always be some part of it – [inaudible] basically some judgment, human eyes looking at it [inaudible] too close [inaudible] be delegated to two different entities.

Let's be honest. We have nothing to do on that as Integration Panel. That is a multiple step process. Similarities should really be taken, to some degree, conservative. You want to capture the ones that are so much in your face. You have to put a limit on those. Beyond that, it's really up to the further process to do the determination.

SARMAD HUSSAIN: Thank you. And thank you all for staying longer than the scheduled time and participating. Let's thank the panelists and then conclude the session. Thank you.

[END OF TRANSCRIPTION]