KOBE – ICANN GDD: IDN RZ-LGR Workshop
Wednesday, March 13, 2019 – 15:15 to 16:45 JST
ICANN64 | Kobe, Japan

SARMAD HUSSAIN: Thank you very much for joining us at the IDN Root Zone LGR Workshop on 13th of March 2019. In our program today we have a series of presentations. We will start with a presentation on variant considerations by Michel Suignard who is a member of the Integration Panel. It will be followed by an update by the Root Zone Study Group which will be presented by the Chair of the study group, Dennis Tanaka. After that, we will have three community updates. We'll have an update from the Chinese Generation Panel, presented by Wei Wang and Kenny Huang, who are the Co-Chairs of the Chinese Generation Panel. We will have an update from the Japanese Generation Panel by Yoshira Yoneya who is a member of the Japanese Generation Panel, and an update from our Korean Generation Panel by Professor Dongman Lee, who is a member of the Korean GP, followed by a question and answer session. So, let us start with the first presentation, over to Michel Suignard for our presentation on variant considerations.

MICHEL SUIGNARD: Okay, good afternoon. I hope you were not too shaken by the earthquake. I for sure felt it on the 11th floor when it happened. Anyway, it was –

UNIDENTIFIED MALE: [off mic].

MICHEL SUIGNARD: Yeah, anyway, so I was on the 11th floor when it happened so I did feel it a bit more. Anyway, okay, moving on. So, I mean, a see a lot of familiar faces here, and so I think a lot of what I'm going to say is going to be familiar to most of you, so I'll try to go quickly through some of those concepts. Still, the very first point that we want to say about variants, the main thing is they are exchangeable. It's not a matter about it being identical, it's exchangeable. It's basically that users will accept one for the other even though independently there are multiple reasons why things can be exchanged, and I'm going to go more into detail about what we mean by that.

Obviously, the variant has to be symmetric and transitive. It bears repeating that we see in LGRs sometimes issues, especially even in context. I mean, in variants you have to make sure that you express both ways, including for example if you have a context on the variant, you have to make sure that the same context exists

on the others, other way. On transitive, we see that issue in A ~ B and B ~ C. That means that A ~ C are also related. So, this is the concept we have been seeing for a long time, so.

This is probably a bit new for some of you. I wanted to kind of create, put an exhaustive list of what we consider variants. So, it can be semantic, it can be phonemic, they can be functionally equivalent or visual identical because everyone can then concentrate on the last one, especially outside of the CJK universe but in fact we have a lot of different cases.

Semantic is when you receive something that is very present in the CJK universe, mostly through simplified and traditional, but it's not the only case here. You also have the case where multiple traditionals can be semantically recurrent, or even multiple simplified, although that's much less common.

Arabic, we've seen a case where we have semantic variants, where you have orthography variants, and I'm showing just some examples. On phonemic or phonetic variants, this is one that we have already implemented on the root zone. That's with Ethiopic, where one of the writing systems that is Ethiopic script, Amharic has a lot of phonetic equivalence. In fact, it's very large because, I think, in fact, about 20% of the repertoire, and so it's extremely large. And 20% [on cities]. It's kind of interesting with such a large set of variants. It only affects about 1% of the [copies]. So, it just

kind of gives some idea that, even when you have a fairly large variant system, the effect of blocking each other is, in fact, fairly limited when you talk about labels. This will also come in maybe later on the presentation.

And then we have the situation of what I call "functionally equivalent" or "functionally exchangeable." There are not many cases of that, but I've found one that is probably useful. It's in Latin, "oe" which is a root ligature is a common occurrence in French, and for a lot of reasons I have to do more with the IT issues. When the French writing system was implemented on computers, the "oe" was kind of left out of the main Latin one, the repertoire, and as a consequence a lot of the French users started to use the sequence to represent a word that should be written with the "oe" ligature. So, now on registration, it is very common from a French perception to perceive that the "oe" as a sequence is equivalent to the "oe" as a ligature.

I mean, it can be debated. I mean, we see the Latin GP as the decision to be made about what to do about that. Is it to consider it, or not? But the list is a good example where, in fact, IT deficiency in the old days created a variant which should not have existed to some degree.

And then you have the case of identical, and we took a case that is totally obvious in Latin where you have two letters that have

exactly the same lowercase form like the schwa on the "ə" although if I remember, well, the uppercase looks different. But. Because at the end of the day, with the lowercase, the representation in that case is exactly identical, what we call homoglyph or homograph.

In Arabic there are also some of those cases where, even if user-related form is distinct, they may share a lot of what position they're from when they are not – you know, it is related but it's a medial final or initial?

So, that kind of describes the different variant types with the database. So, typically, you can specify, depending on the scripts, the variants in one category or in multiple. As you can see, Arabic has multiple types of variants. It can either be semantic or visually identical. It's not just only one. We can see that happening in other writing systems as well.

Obviously, an important point is also that a code point variant is defined a as code point, but they do affect variant labels. What we're really trying to address is variants labels and not variant code points, right? Code points are just a mechanism to reach the variant label issue and we see that is happening when we apply the distribution on the LGR. So, the LGR is a set of things. You first define there is a variants at the code point level, and then you take action on them. At the end of LGR, you have a set of actions,

and then you have a disposition, and then this is a label that goes through that's LGR process, is discarded or is valid or invalid, and you can also create an allocatable version, or blocked, obviously.

Users interact with variant levels and only interact with variant code points. The fact that two code points are variants doesn't matter. This only happening in the labels. For example, combining marks. We will find a lot of variant systems have combining marks that are extremely confusable. In fact, identicals, but if they cannot be attached to a base character, it doesn't really matter because – so you need at least one base character to basically make the combining marks that a variant which are to be basically meaningful.

We have quite a few examples like that in the writing system where there are a lot of, [a few matters], that are confusable, but not when they're written with the consonant.

Obviously, code points have no context other than the label. I mean, you don't have a full context, you don't have a sentence, you don't even know which writing system you have, you just know the script. The information you have at the root level is a script. It's not language. It's not a specific writing system. So, that also has brought some variant definitions that you'll not necessarily expect, like Devanagari, many languages or writing systems using the Devanagari script, for many of them, they don't

expect Nukta behind a vowel, and so but see we have to include that because we see for some users it's very unexpected to have a Nukta.

So, they made, in fact, a variant between the vowel, with the Nukta and without the Nukta, although you would think that somebody who knows that word would say, oh, those are different, but for some users of the script, they are not, so we have to – the decision by the GP was to make a variant between the Nukta with the character or the character alone.

So, then we have the situation where we have people we'd consider mostly In-Repertoire variants because that is what is more familiar to them, but that's not the whole story. You also have to look and see. Is the root being shared with us? We have to think of the cross-script variants.

This morning, for example, Sarmad showed the case of Cyrillic and Arabic and that is totally, you can write it in both Latin and Cyrillic, so clearly GPs have to investigate all In-Repertoire, so Out-of-Repertoire.

So, some rules here. We obviously expect the GP must—and that's a big must—investigate all In-Repertoire and declare all of them in the LGR. I mean, that's kind of a given, but it bears repeating.

Out-of-Repertoire is a bit more of a mixed message. We do expect GPs to investigate at least the scripts that are related to their own script. So, we see, for example, a Latin GP would obviously look at Cyrillic and Greek, as the obvious case, and some were not necessarily obvious, at first approach, but like Armenian became, in fact, through some harmonization of glyph and finding much more induced to variants, but you see we don't expect, necessarily, all the GPs to look at everything.

Some of the repertoire are fairly large, if you think of, Ethiopic or even more so CJK, although in that case it would be extremely not to be expected that you would have a lot of variants between CJK and anything else, maybe with the exception of Hanja, I mean Hangual, sorry.

Anyway, so the thing is, so Out-of-Repertoire variants can be added for integration. It's not, if you have not found variants for your own study, the IP has a mandate to kind of check that and go deeper, and that is of various scripts to kind of detect them and add them. Even if they are not on the original LGR, we can still add them as part of the integration.

What we would obviously never do is to add in-script variants as a result of this integration, at least. There is some exception on all these mechanisms, which is for shared scripts scenarios. In that case, the shared case is really CJK in this case, and the story is a

bit more complicated here. In fact, we have to go through multiple train of thoughts and iteration to kind of come to some degree of decision on this. We want to make sure that, in fact, when you have a shared scripts situation, the original variants, submitted by the original owner of the LGR keep their variants, and the main thing is really to harmonize the mapping types.

There's a big difference between when you share scripts, you can still keep – you have the same variants set because the variants set at the end is very – it is common for the variants sets on the shared script, but the mapping types, and that his how you dispose of the variants, for example, if you declare them either to be blocked, or locatable, or some mitigation to the remitted number of the locatable, it is totally your own. So, that belongs to each individual LGR because the common LGR only makes everything blocked, but in each of the LGRs, you can create your own rules.

The other thing that I would also say is that you cannot – when you create cross-script variants, if you add – you cannot basically create in-script variants in another script without at least, kind of talking about it or making sure that they know about. Again, that's except for the short script scenario. That's a bit separate.

We had that situation, by the way, between Latin and Cyrillic, where we had a situation where the Latin GP created an

occurrence between two sort of "y and the character looked like a "y" if I remember well, and then that did introduce an in-script equivalence in Cyrillic and that was done with, obviously, cooperation between the two GPs, but you need that cooperation between the two GPs.

Obviously, it goes without saying that you cannot create a script variants that would create a variants between ASCII letters, or a sequence of ASCII letters. You could have, in fact, some scenarios, where that happens, but that is not possible. It would never be allowed to create variants between ASCII letters. It's just not possible.

So, that is another case we change our minds to some degree on the IP. We thought that having single – if you have two scripts that only share one shape, it's not really worth creating a variant for that because you know that – again, we know that variants are done as labels and not as code points. So, we thought the only case you could do spoofing was by repeating that letter many times because you cannot change anything else, and so we thought what would create the multiple "ooo" a domain when we should have known better because it does exist. The triple o (ooo) is, in fact, delegated domain.

So, that means that we have to make sure that we are blocking all the scripts that have something that looks like an "o" in them, and

so we have in fact to add – for example, so the list here is the "ooo." We cannot do that, and then we look around on the MSR, all the scripts that have something that looks like an "o" and we found, we see the usual suspects, like the Latin, Cyrillic and Greek, and even the Armenian, but then we discover that, oh, yeah, Malayalam and Myanmar also have the "o," a discrepancy enough in its own iteration. Obviously, if we look side by side to the Latin, you could say that Malayalam is a bit too big, the Myanmar is a tiny bit too small, but if they are just on their own, you don't know.

And the same for some other variants, and we are not totally free done on that, but like the "c" that we see is an open circle, and we see and again expect Latin and Cyrillic to be there, but the surprising one is Myanmar. You can do, in fact, that "coco" is perfectly representable, and it can be represented in Myanmar. It obviously doesn't mean anything near the meaning of "cocoa," if whatever would be that in Latin or Cyrillic, but those are the cases where the vertical bar is a bit more rare. I mean, obviously, you can find that in Latin and Cyrillic, again, and I try a bit, oh, yeah, I don't think that we get to a point of being confusable, but I wanted to put it on this slide at least as – or, you could find more.

I mean, that's basically the case of the simple shapes can be, in fact, confusable because they don't really come to very much of the semantic or meaning of the universal writing system. They

kind of are really a bit poor innovation on graphic elimination. So, it's simple as the graphic, the glyph is, it's easier to make a spoofing attack. The more complex they are, it's a bit more difficult because they do convey information that most users would distinguish.

So, yeah, we are in discussion with especially the CJK panels about the visual, what constitutes visual variants, and we use, we avoid very carefully to use, to say "visual similarity" because that is a term which, by our own procedure, is really out of scope for us. We say that this is outside, it should be done by a mechanism, beyond the LGR that involves, basically, some human eye or some judgment to judge if this is creating an issue, and this typically is not done automatically. Although, again, I mean, even for variants based on visual similarities, as a procedure, we recognize that there is some gray zone, as written on this text, where if there is way to decrease a load of the visual similarity panels, in fact, you have to make sure that you do the best effort to minimize that kind of work.

Obviously, we have to, again, remember that the IDN labels are not really normal text, more restrictive. They're not meant to represent words, necessarily. They can be just expressions, or they can be a subset of the writing system. They don't have to be fully representative. At the same time, it's kind of ironic that they also have more flexibility because you can create things that

don't make any sense that have totally wrong spellings, as long as they are well formed, and in fact these are fine. Many writing systems allow acronyms, for example, don't make any sense on their own, unless they're related to a brand or you can basically decompose the acronym into a meaningful componence.

Again, the procedure on the IAB, the Internet Architecture Board, is really pushing us to be conservative and that has affected, for example, like for some characters they look almost like punctuation, so we had to be a bit, we have been pushing back on punctuation-like characters. And, again, to be acting on the side of caution about that so that they visually don't occur.

I like to use the example here. We're in Japan. So, I know this is a bit controversial, so I will not go too much on it, but still that's taberu, a word in Japanese. The verb says "to eat" and the normal way to write that is the one on top, which is Hirta," you know, the Kanji "ta" followed by "beru," the irregular, and the second form, the second one, the only thing I changed was too the middle one, replacing the Katakana "ber" with the same. So, I'm not answering the question, but I'm making the question can they be exchanged or not? And that's something that at some point that the Japanese GP will have to at least analyze. We are not even, at this point, giving an answer, but they do have to analyze that case to consider and at least have to, in their documents say, oh, we looked at that, but we didn't think it was confusable, or the other

way. The IPs at least want to, when they have a case like that, they need to be exposed. We are not at this point prejudging what should be the solution, but they have to be exposed.

Again, this is also the case where we don't want suddenly the GPs are facing thousands and thousands of characters to go in a big complicated story, I mean, the research on this. You can look at a simple case, a simple, things have to be reasonably simple. So, the user could be confused and could exchange one for the other.

On the other side, again, and very often that's the case, is the characters are extremely complicated, and there's only a very subtle difference, and you have to also consider that case. There's only like one tiny piece, a bit of a stroke that is different in a character that has 40 or 50 strokes, and that also is a concern because most people won't see the difference.

Allocatable versus blocked. The IP has no issue of having as many blocked as you want. Typically, a blocked variant can be as many as you want. In fact, you can have an LGR with a lot of blocked variants, and we have seen from experience that, in fact, you can have a very large number of block variants without impacting significantly the numbers of labels you can create. We've seen that, in fact, in the example I was giving before with the Ethiopic.

On the opposite side, when you do allocatable, you really have to, in all cases, and we are not being – we want any script, I mean,

any LGR that created an allocatable to have a mitigation system to limit the number of allocatable variants. So far, all of them have done so. Arabic uses all label evaluation rules to limit permutation because you understand that if you don't do this thing, if you have a very long label, you can have, I mean, a very large number of allocatable variants. You could have hundreds, possibly, or even more if you make a very long label. So, we have to do something about that.

In the Chinese, on the draft version, we have – in fact, based on the inputs from the IP, the Chinese GP came up with a mechanism to limit the number allocatable by typically having one original, one traditional and one simplified because there was originally some variants where you had multiple traditional, and even in some cases multiple simplified, and so a scheme was kind of created to limit that, to have kind of preferred traditional, if you want, or preferred simplified. We created a solution that is managing a number of permutations.

And the same thing, we have had recently a case where a GP submitted a version with a SHRI variant which is fairly common there. Originally, the LGR did not have a mitigation for those at the time the GP came up with the mechanisms that, if your label contained one version of the variant, it has to be consistent through the label. So, if you use the one form as a label, you can only use that form for the full label. So, that means you only have

– you know, if you have a label that contains one of those, or multiple, you only have one variant for the full label.

So, that's what we want to see. We want to see these kind of things. Basically, if the LGR presents an allocatable without some mitigation for the allocatable, the IP would for sure send that back and ask please create a mitigation. You have a choice. You can expect different ways. We, in fact, also use a slightly different ways of doing this mitigation. So, you can either try to create something on your own, or you can also work with the IP to get a solution for this.

This is an example, and I'm not going to go into detail. It's a bit complicated. It's more like for further study. It's basically showing that in some cases variants have their own context. We have typically contexts we use on code points in LGR, some especially in the [inaudible] LGR we use a lot of those contexts to create, to enforce well-formed roots on the writing system, like for example, what can follow or proceed a given character.

So, we have a lot of those rules, but you can also put them on variants. Variants can also have a context, and that's in fact pretty powerful. In some cases, like for example, when we have what we call null variants, so you have a sequence that mapped to characters that only contain one of them. So, in essence, you're creating null variants for one of them. And that will still create a

situation where you have to make sure that the context for the variant exists in both ways, even if it's not needed for one way. So, you still have to do that, even if it's not obvious.

The other thing also you have to do, I mentioned, or I'm not sure if I mentioned that here, but sometimes you have a context on the variants, and you have another context on the code point. In that case, as well, the two contexts need to be consistent. So, you cannot create a context on the code point that is not compatible to the context on the variants side.

Again, we had that situation, in some cases, and I think it was in Malayalam where we had this situation, exactly, where we had to kind of fix. We're still kind of working on it, by the way. So, we have to be consistent and I encourage you, if you want to understand, or go deeper on this thing, to look at the Devanagari LGR and it has been made public. So, you can look at it and go through the gory details of the XML on the documentation to understand this thing.

Okay, I think that concludes my presentation and I will be open to questions later. These are some references that you can look in more detail on this thing. And, again, the LGRs, in fact, are a good source of information. You can look at all of them and go through the different things, the different details. They're really a good source of information for many of those. Okay, thank you.

SARMAD HUSSAIN: Thank you, Michel. We'll continue with the update from the Root Zone LGR Study Group. So, over to Dennis.

DENNIS TANAKA: Hello, Dennis, for the record. So, Root Zone LGR Study Group. This is the agenda for today. Just as a way of background, I think many, if not all of you, know that there's a Root Zone LGR available through the LGR Procedures.

So, the LGR procedures, the Root Zone LGR is a work in progress of the LGR Procedures, which will help us validate labels and calculate the variants, and their dispositions, but that's only one, that's a tool. Now, this tool needs to be incorporated in the corresponding policies, such that it can be used to, actually, fulfill its purpose which is to validate TLDs. Therefore, this study group is looking at a way on how to harmonize the use by gTLDs and ccTLDs moving forward, and we are looking at recommendations from a technical standpoint.

This is a composition of the study group. Some of them are here with me today. As far as the scope of work, these are the areas where we felt that we needed to address. They range from looking at the users, who will use these rules in the LGR. One thing is obvious, the TLD applicants, but there are other users such as

generation panels, integration panels themselves and the ICANN community, at large.

What does it do? We have to be very clear, from a technical standpoint, that the Root Zone LGR is a tool for syntax validation and the calculation of the variants. That's basically a very limited scope as to what the Root Zone LGR does.

Why is it important that we have a Root Zone LGR? I mean, in the past, the way that IDN TLDs were applied for is based on their own applicant view or using an IDN table which was widely available, but a different IDN table for that matter. So, the reasoning behind having a one, single authoritative Root Zone LGR is to have a consistent predictable way to validate all TLDs that are applied for throughout the ICANN process.

Other items that we are looking at is when do we apply it, for example, early in the process, as opposed to as a way to validate and potentially replace some of the previous processes such as a DNS stability review, for example, where a Root Zone LGR would now be one tool that can help the DNS stability review process, for that matter.

Another area that we look at is, where do we find it? Our Root Zone LGR, I mean, when we talk about the Root Zone LGR it's the XML authoritative file that is the result of all of the integration of all of the script proposals. So, what is going to be found? For

example, one idea is to have IANA serve as a host of the authoritative files so that any implementer can go to IANA, find it and make their own Root Zone LGR tool.

We have other considerations, as well, looking at other security and instability issues, or the movement of abandoned states, when the Root Zone LGR is updated from version to version. What is not in the scope is clear. For example, semantic validation.

Now, what is treated an IDN and ccTLD or what is a geo-name, brand, community, TLD, etcetera, that's for policy to define their own criteria and define whether they pass or not, right? The Root Zone LGR is merely a technical, a syntax validation and calculation of the variant labels, regardless of whether the TLDs are going to be applying for as geo-name, brand, ccTLD and what have you.

Also, what is not in the scope of the study group is to come up with a number of allocatable variant TLDs. Again, the Root Zone LGR will calculate the variants depending on the code, and it's going to be up to policy to define that number.

Also, what is not in discussed in this study group is how to process the TLD applications whose script is not yet supported by the Root Zone LGR, right? We know, for example, Root Zone LGR v2 has a number of scripts and let's just, for the sake of the example, the round of new utility opens tomorrow, so what are going to be

the TLDs that aren't going to be processed by the Root Zone LGR? It's only those that are already now integrated, right?

So, what we are doing in the study group is to look at the different considerations that subsequent policy will need to take into account in order to address this problem, but the study group, in no way, shape or form, will discuss as to the how that this needs to be done.

So, next, we're going to look at what is the current state of the Recommendation Study Group. This is in no particular order, but these are some of the areas that we are discussing.

So, starting from the top, personal LGR is meant for all TLDs. I know the Root Zone LGR started as an IDN Root Zone LGR, but it's become obvious that the ASCII is a subset of the whole Latin script. Therefore, it is logical that all TLDs including ASCII, be subject to Root Zone LGR processing, especially because there are cross-script variants across Latin, Cyrillic, Armenian and Greek.

The second bullet is for scripts or writing systems that are integrated into the Root Zone LGR, that Root Zone LGR is the sole authoritative source to validate and calculate variants, right? If policy adopts the Root Zone LGR as a tool to do it, it has to be the sole authoritative source so that all processing is consistent and predictable.

The next one is, if number two is true, then policy cannot just overturn the calculation of the Root Zone LGR tool, meaning if the tool, if the Root Zone LGR declares, or terms that a specific string is invalid, then policy cannot say now it's valid because of some exception, right?

So, it's an "all or nothing" kind of a scenario, but we recognize that there are going to be cases because ICANN cannot control the output of or the formation of generation panels in order to work on those scripts. So, there may be, potentially, possibly scripts that are not going to be incorporated in the Root Zone LGR in time for any application process for the TLDs. Therefore, while we do not define, or we do not come out with any recommendations, we are just saying we are letting policy define that and make a decision whether it's advisable to process any application or apply for a TLD outside a Root Zone LGR.

Next, we look at the changes into the components in the Root Zone LGR and how it affects existing TLDs. Basically, what we want here is to preserve the stability of the root zone, and therefore any versions of the Root Zone LGR should be, if at all possible, backward compatible with previous versions, so that existing TLDs are not invalidated by the newer version of the LGR.

Next, there should be one, and only, authoritative source for the Root Zone LGR XML file, and as an example, I say it's IANA, in the

study group, we think IANA, it's a good candidate because today it serves also as a host for the second-level IDN tables for a registry operator. So, IANA feels like a good place to have the authoritative source for the IDN table for the root zone.

But the XML, it's just an XML, it's not really human-readable, it's not meant to be used by humans, so we need some kind of implementation of it and most of you know the Root Zone LGR tool that is provided by ICANN. There's an opensource, I believe, right? It's still available for anyone to take it up and implement the same tool or a different tool with all the features so that it's widespread, but we think that at least there should be, as a community service, one implementation that other community members can use, can use it right away.

Other topics is the allocatable variant labels. I said that it was not in the scope, setting a number, but we are just looking at endorsing or considering endorsing what SAC 060 report says about the number of allocatable variants which should be as small as possible, and so that's the LGR procedure.

And, last but not least, what about the other things that we need to look at? It's an appeal process so that, for example, a TLD applicant disagrees with the calculation of an applied-for TLD, and what is the process by which this TLD applicant can avail himself to make the case, right?

Again, this is a policy discussion, but what we as a technical study group can say is that any solution needs to conform to the LGR procedure. So, let it be a change in the repertoire, adding a new code point, or adding or removing a variant code point, or a whole level evaluation role, it needs to conform to the LGR procedures, which basically means the generation panel needs to review that such change and it needs to be integrated by the integration panel and then adopted when the new Root Zone LGR is released.

And that's if for me. Any questions?

SARMAD HUSSAIN: Yoneya, you have a question?

YOSHIRA YONEYA: So, I have a question for both Michel and Tanaka. So, for Michel, I couldn't understand on this right-hand side, slide 11, that you said only variants. I was confused that variants means code point variants or variant label because on this right, slide 11, you showed the variant label. So, the variant relation between Hiragana and the Katakana we don't think it is a variant, but this is visual seniority over the label, so I was very confused with the variants you used for explaining this.

MICHEL SUIGNARD: I will try to explain that. I mean, obviously, you define code points. You define variants and code points, and that's the only thing we have in LGR. We don't define label variants, if you want, we just define when you create LGR. What you did was basically code points and some cases sequences. We use sequences in some cases, and for that reason, and not for variants in fact. So, you define it at the code point level, but when you apply the LGR, you apply it, not to code points, but you apply it to a sequence of code points, basically a label.

So, when you do apply for processing you have, for example, and applied-for label which is basically a sequence of characters, and then you compute all the variants that exist or are possible for that label. That's the only thing I mean.

So, let's say, the first one, the Taberu, is applied for, then there is the LGR where you will feed that to the LGR system, depending on what code point variant you created in the LGR, the LGR will lead to the set of labels that correspond, right, to the soft label, and those would be either block variants or allocatable variants under which it would not be a code point, it would be a sequence. Again, it would be a label that looks like, or is considered by a user to be exchangeable with the original one. That's the only thing I meant.

So, you see, the LGR only deals with code points for variants. That's the only level we know in the LGR, but so the way that you process them, the way that you use them is to use them on labels.

I mean, sorry if I made that unclear because that's the new, that doesn't mean it's the case forever. I'm not introducing anything, a new concept yet. I'm just trying to – maybe I misinterpreted the procedure, or misread it, but I didn't want to say anything different than what is on the procedure at this point.

SARMAD HUSSAIN: So, we'll take one more question or comment and then we'll move to the next presentations, and then we'll come back to questions and answers at the end. So, we have Edmon, first, in the queue, but if you want to –

UNIDENTIFIED MALE: We only have one now.

SARMAD HUSSAIN: Okay, sure so.

YOSHIRA YONEYA: Before going to the next, I have another question to Tanaka, so can I finish my question?

SARMAD HUSSAIN: Is your question related to Michel, first, or?

DONGMAN LEE: This is Dongman Lee from KGP. Along with the Yoneya's question, in the slide of, it was number 6 of Michel's slide, number 6? Yeah, he actually describes the types of variants and the slide number 11's example, I don't know which category his example actually is belonging to because there are no visual similarity variants, so similar variants.

MICHEL SUIGNARD: Visually identical I think is –

DONGMAN LEE: But those two things are not visually identical, even I'm not a Japanese person, but the Hiragana ba versus the Katakana ba is, there is no couverture. If you look at the fonts, the ba, the upper one has a couverture, at the end, echosign. So, who people who actually know two characters, character sets, they know because when they actually write the character, they are not the visually – I don't think those two things are really – belong to – so, that's I would like to –

MICHEL SUIGNARD: Again, sorry to interrupt you, I understand, Again, the LGRs, the original LGR, the original LGR is shared by everyone. It's not just for specialists and I use that example very carefully. I didn't take any random example. I used that one because it's probably the more visually identical set you can find in Japanese, frankly, one of the – and depending on the font, I agree that if you take a serif font, the difference is obvious, and it obviously has different uses. Nobody is denying that. It's just that if you – in a lot of fonts, in these modern fonts, especially, in Japanese, if outside of the context you can – I mean, if you look side by side, the one thing, again, for variants that you have to consider, yeah, I should stop, otherwise we'll be on that forever, but you have to look – labels are evaluated typically out of context.

So, remember that also. They are not side by side. You are not, this is luxury you don't have, typically. They're totally basically evaluated in isolation.

SARMAD HUSSAIN: Thank you, we'll take Yoneya's question, too, for Dennis and then we'll move on.

YOSHIRA YONEYA: So, mine, this is Yoshira Yoneya. My next question to Tanaka is the purpose, the recommendation is to whom – I, so, who reads these recommendations?

DENNIS TANAKA: Thank you. So, the recommendations, so the study group was convened by the request of the ICANN Board. So, we are going to submit this back to the ICANN Board, and they will likely, as the SO organization, I mean the supporting organization, GNSO, they are taking that into account in their own processes.

YOSHIRA YONEYA: Okay, thank you.

SARMAD HUSSAIN: Thank you. So, we move on to the Community Generation Panel Updates. Can we, and the first update is on the Chinese Generation Panel, so over Wei Wang and Kenny Wang.

WEI WANG: Good afternoon everyone. It's a pleasure to give the updates of CGP on behalf of the panel. First, let's take a review of the CGP work. Actually, the whole panel was seated in September 2014, almost five years ago. We have 23 experts from 10 countries include China mainland, Taiwan, Hong Kong, Macau, Singapore,

Malaysia, as well as the experts from London and the United States.

Also, it's very – we are happy to have Edmon Chang, the CEO of DotAsia, and the co-chair of UASG to be the contact and advisor appointed by them. Also, we have the frequent interaction with the JGP and KGP because we share lot of Hanji characters in different communities, and we are happy to have the IP to give us suggestions during the whole process.

Well, let's take a look at this graph. There are many bubbles in this graph, as you may see that we were seated in 2014, and now it's 2019. We already generated 12 versions of our document, revised little by little, and according to the feedback from the J and K, or as well as the feedback from the IP.

The strategic coordination is very important to the CGP work because you may see that the Chinese characters are not only used in mainland and other Chinese language communities, but as well as in the Korean language community and the Japanese language community.

The current CGP repertoire is basically a collection of the CDNC repertoire and dotAsia repertoire. We removed some, a couple, dozens of characters we imported from he CGP and other tables in the last version. The biggest change currently is that the CDNC

table includes two characters from the dotAsia, which decrease the dotAsia unique characters from 124 to 122.

Actually, the repertoire itself, it doesn't change. The CJK coordination, as you may see there, there are – I'm sorry, I didn't put an example on the slide, but for the Chinese variant, the definition is that the characters represent the pronunciation and semantic meanings which could be exchangeable or defined as a variant, and for every single Chinese character it's supposed to be a simplified form and a traditional form, and perhaps with some other rarer forms, the simplified one and the traditional one are supposed to be allocatable. The other ones are supposed to be blocked.

So, generally, I think 99% of the CDNC table and the dotAsia table, they are a disposition on allocatable, and simplified allocatable ones are the same. Except for 109 characters, we have different variant mapping settings for those 109 characters, and to coordinate – so, the first job for us today is to do some internal coordination between the CDNC and the dotAsia tables.

We invited experts to form as CGP internal review panel and generate the CGP variant admins. Before our coordination was K. So, after that was carried out, we conducted the coordination with J and K, the external coordination. J is very open, and we decided to import all of the variant settings from C, which mean J

itself didn't raise any objection to our current variant settings. However, from the J's perspective, the Korean community raised like 258 variant groups which are deemed as not their variant from the Korean perspective.

So, we did a couple of runs, coordination from 2016 to 2017, and reached a consensus which divided the 258 groups into two parts. Among the groups, only 146 groups will be split into independent characters, involving 445 variant mapping entries. However, I have to admit that we reached a consensus, but it doesn't mean the Chinese community think they are unrelated. Still, the Chinese community thinks it's just submitting to the meaning and are exchangeable. The usage should be exchangeable.

So, the next job is that we have to reconsider how to apply this consensus or compromise from a very conservative perspective. We did some analysis on the application of those characters at the second level. We found that, at the second level, in the database registry, the registration data of .cn, .tw., .hk and 网址, which is a new gTLD, at the second level there are about 117 groups which have never been registered at the second level. So, which means it might be safe to split them, but I have to admit that historically there is no registration. It doesn't mean that in the future there is no registration requirement. So, it must be, I think for the CGP we should be very careful to consider how to apply this consensus in the LGR.

Another thing is the visual similarity. According to the confusable tables at the website of Unicode, there are some han character-to-han character pairs with visual similarity. We analyzed these six pairs and think the two pairs are not very similar, so will insist on separating them, but the other four we probably will treat them as visual similarity, and we are thinking about proposing some disposition to handle, to address the issue.

But, as well, we noticed that there are also visual similarity issues for K and for J. If it's possible, I personally would like to see that, all of the three parties would take the same – we took the same solution, design the same solution to address this issue.

So, for the visual similarity issue and the unacceptable variant disposition from the K, I think there were two options for us. One option is we make our decision and apply that into the LGR, to make the effect of the LGR resume directly. The second option is that we probably could apply it in the post-evaluation process, which means we don't need to make it a full, automatic algorithm which will block the unexpected one and generate only the allocatable one, which is kind of – I think it's maybe a little bit aggressive or risky because it excludes the possibility for the community members to apply the potential labels. So, another option is to put in a post post-process.

I think that the next step for us is to synch up CDNC table and the dotAsia table with the CGP. As I said, and as I mentioned, now, the CGP repertoire is only, it's a collection of dotAsia and CDNC. The only difference is the 109 characters. So, I think it might be time for CDNC and dotAsia to synch up with the CGP to make the top level and the second level share the same repertoire and the variant mapping settings.

A second is we need to reconsider if there is a more conservative and secure solution to the C and K coordination, but the third is thing about the variant similarity. If we, the C have to decide to address the issue in the LGR to design some drawers in the LGR, or to leave the problem, the issue to the post-string evaluation process, or to wait for J and K to propose their solution and try to make a universal solution work for the three parties. Thanks, that's all.

SARMAD HUSSAIN: Thank you, we'll move on to the next update on the Japanese Generation Panel by Yoshira Yoneya.

YOSHIRA YONEYA: Thank you, this is Yoshira Yoneya from JGP. I'll briefly explain about the JGP update. The status of the JGP is here. So, we started our work in 2014 and we defined the requirements and

basic framework of the Japanese LGR in 2015 and we decided – oh, I'm sorry, we made almost a final proposal, but still studying, seeing the visual similarity is raised, so that we still are waiting for the distribution of visual similarity or visually identicalness. And this is a member of our panel, and this is an overview of the Japanese LGR. So, we use three different scripts. One is Kanji, and Kanji is the same as Han or Hangul, and the Hiragana and Katakana.

We selected our [reporter] from JIS. This is the Japanese Industrial Standards level-1 and level-2, used widely in Japan, and those include over 6,000 characters, most of them Kanji.

And variants. So, we decided we do not define our own variant, but we import variants from CGP and KJP and if their definition fits. And, at this moment, we do not define a variant for Hiragana and Katakana, but as I said, the visual similarity issue is still remaining. And for the WLE, we defined a very small set of WLE, which to reduce allocatable variant levels.

So, this graph shows the position each C/J/K GPs, but this is already described by Wei Wang, so I skip this, and this is how we reduce the allocatable variant labels in WLE. So, as Michel said, if the character has a variant, then the label could be increased much more by how many variants the regular includes.

So, as I said, JGP imports variants from C and K. So, in Japanese labels we don't have many variant labels. So, to reduce our allocatable variant labels the JPG and IP discussed, and now we are thinking about daily-use Kanji, and it is the smallest in our repertoire. It is about 2,000 Kanji, and using that daily use set we can reduce the variant levels, allocate the variant levels to the very small number, so that really at this moment the JGP and IP are really on this thought. And we are now, ["we" meaning] the C/J/K and IP, are discussing how to handle the visual similarity or the visual identicalness.

So, initially, C and J and K thought that the variants, the characters which had the same meaning and the same pronunciation, but the visual shape is not the matter. However, ICANN and IP started to request C/J/K GPs to handle characters with visual similarities in the Root LGR. So, we discussed more than one year and still have not concluded.

So, end of this January, C and J and K sent a correspondence to Göran Marby and we received a response from Cyrus this February, and this is still not resolved.

So, this slide shows the abstract of the concept. So, just we think that the visual similarity should be resolved outside of LGR and the background is for your reference, and so I won't explain briefly at this moment. So, this is all from the JGP update.

SARMAD HUSSAIN: Thank you, Yoneya. And let's move to the last update by the Korean GP and I request Professor Lee to do the update.

DONGMAN LEE: Okay, my name is Dongman Lee. I'm a member of the Korean GP. So, it is a combination or [overview] of what the KGP, so far, has done since 2015, and so I'll just quickly go through the overview of what character sets we include in Korean, the variant set, and I'll focus on more the latest update on our activities. So, as you may know, the K-LGR covers not only Hangul and also Hanga. Hanga is the Korean pronunciation of Hanzi or Kanji. So, the language is not only just used in South Korea, but also North Korea, and all the Korean descendents, or the people who actually used the Korean language in all of the world.

So, the KGP members consist of quite diverse areas, not only technical experts, but we have also linguistic experts, Hangul plus Hanja, and these people actually really have helped in the coordination with the C and J for the coming of the unified variant set, and we have registry and registrar, and policymakers, as well.

So, since 2015 at the beginning of the formation, and to around the end of 2017, our major activities pretty much focused on coming up with the unified variant set with Chinese, the

generation panel. So, as a record, Wang Wei already explained the variant set, but as far as the Korean LGR is concerned, we have about 152 variant groups, which include the 223 Korean Hanja. And, anyway, over the whole Hanja studies there are 4758 characters, and Hangul we have 111,172 syllables defined in a Unicode set. So, variant groups that compose these Hanja characters are also Hangul syllables, but we're going to talk about that later.

So, I should mention these. So, we published the first version of the LGR proposal at the end of 2017, and we had the various, the public comments, but the main idea was mostly the young generation, they actually show some resistance to include the Hanja characters. Since then, in 2018 from April, we had a series of meetings with the people who made the public comments and also invited other people, the players in the Korean Internet community, and in a summary, we understood that their intention, their main intention was not agreeing on the mixture of the Hangul and the Hanja in the same label. So, basically, they understood. Through the series of meetings, they also understood the old generations are very fond of using Hanja characters as well.

So, in the Korean community, we cannot simply say Hangul-only or Hanja-only. No, but this is kind of been using the last, you know, almost, well over 500 years. So, we cannot simply avoid

one or the other. So, we happily concluded at this point the consensus on Hangul-only or Hanja-only labels, but we don't yet have the consensus on mixed labels.

Also, in the previous slide the reason why we mentioned this person is that this person really, with great enthusiasm, line by line, reviewed and gave us comments, but happily we resolved all the issues. So, as based on the last time, the last Barcelona meeting, IP suggested to us to include the summary of those comments and how we resolved, and so we plan to include the next version of the K-LGR, the document, so you will see.

And the visual similarity part, we are pretty much in synch with our Chinese and Japanese friends. So, at the same time, in the version 1 K-LGR document we identified five of the Hangul versus Hanja, the visually similar variants, but there could be more, but principle-wise, as I just said, we are in the same position as the Chinese and Japanese GPs. Thank you.

SARMAD HUSSAIN: Thank you, so we will open the floor for questions and comments. We'll start with a comment online. [Regina]?

[REGINA]: There is one comment on line from Syed Iftikhar Hussain Shah. The comment starts, I appreciate such workshops on IDNs and

ICANN forums to create awareness about IDNs within the global community. The same efforts also can be managed at the regional level, as well. This will help us to create awareness and to resolve the relevant IDN issues. End of comment. Thank you.

SARMAD HUSSAIN: Thank you. Any more comments or questions, Bill?

BILL: Yeah, on Michel's slide 9, if you could pull that up, yeah, the Oriya pretty clearly is not confusable with the Latin "olol" but I wonder if it might not be confusable with the Latin "o.lesi o.lesi" sort of label since the "O" and the .lesi "I" are the same height which is what we see in the Oriya case.

MICHEL SUIGNARD: I mean, these are just examples. I'm not trying to make a policy here. I was just using examples and then every GPs have to make up their mind, and then we look at it. But anyway, even the IGR, if we ever did a decision like that, a policy integration would be subject to public review and so people may object or create a different opinion on that. And, again, I don't want to prescribe here. This was just the examples.

BILL: Understood.

SARMAD HUSSAIN: Edmon?

EDMON CHUNG: So, I have two questions, actually, for Michel. First of all, why not ASCII and to ASCII variants. I mean, ultimately maybe it's not a good ideas, but to throw it out without exploring I think is probably not the right approach right, especially when it goes in a round-about way, right? I think what you're trying to say is this, that a particular ASCII does with a non-ASCII character and then it has another relationship with somebody and round-about comes back to another ASCII character. I think we definitely should explore it, and in terms of the process, but ultimately the decision that's being made is a different thing.

MICHEL SUIGNARD: It's basically very scary if you start to go that way because suddenly you would have to explore the ASCII TLDs and the ccTLDs and everything and to make sure that you don't already have a case that clearly would be exclusive. It would be a situation where, because CCs and TLDs are almost using all the alphabet, that soon you would have two CCs that are excluding each other. I don't think that's really practical.

I mean, it would create a kind of realm that I don't think that we want to enter, in my opinion. But, you know, it's not – IP to some degree doesn't care. It's out of scope for us. But I would say that probably we would have someone else, such as the ICANN board who would come back to us and say are you out of your minds? Anyway, but I think, at least from a default position, we think it's not really reasonable to do so, but overseeing this beyond our scope, so.

EDMON CHUNG: So, my second question is, actually, looking at this and also related to slide 11, I think, the same thing. So, we can just stay here. Is there any reason anybody in this room thinks that the String Similarity Review would not catch these? Is there any plausible reason? Because I think that's a very important question we need to ask, right?

The question is whether, ultimately, we need to decide whether, sometimes, LGR would cater to these issues, and sometimes it won't, but if you look at these, and you say there is no possible reason, plausible, reasonable reason that String Similarity Review would not catch this case, then in cases where it's not appropriate to be included in the LGR, it shouldn't. It should be left for the other processes. I think that's something that we really need to realize and understand. Thank you.

MICHEL SUIGNARD: So, that's a good question, and the answer is, yes, a String Similarity Review Panel will catch this, but I think if you're – I'm assuming you're asking why this is done in the Root Zone LGR, or just that –

EDMON CHUNG: The point is that sometimes it will be done by the LGR and sometimes it might now. These cases, visual similarity or – like visual similarity, basically, some of them it's appropriate to be included in the LGR and some of them may not be, but even it may not be, we shouldn't be worried about it because we have the String Similarity Review. That's the main point.

MICHEL SUIGNARD: Okay, thank you.

SARMAD HUSSAIN: Bill?

BILL: I would just say on that the String Similarity Review would then depend on someone coming up with essentially a table of what is similar enough that the String Similarity Review Panel should cover it because whoever is on that panel isn't going to be familiar

with the entire repertoire across all alphabets, they are just too many characters to think about.

SARMAD HUSSAIN: I'll make one comment and then we'll come back to you. So, one potential disadvantage of that approach, though, was for example having a similar for identical cases, at least the obvious cases in the LGR, is that if a similarity – eventually, it is the Similarity Review Panel which takes care of that. The application has already been submitted which means that it will go into contention and the financial, I guess, investment has been made by the applicant. Whereas, if the obvious cases are potentially caught at the application level through the Root Zone LGR, some of those cases, at least the obvious cases, potentially can be avoided. So, go back, going back to Edmon.

EDMON CHUNG: So, Edmon here, so in response to that, you have to understand the String Similarity Review looks at three things. I think it should be three things. One is the reserved names, one is existing TLDs and one is the other applicants in the same round, applications in the same round. It doesn't need to look into the future things, and what we are talking about is not to block a registration, right, or application, right? It's only when there is an existing TLD and two applications then, and they are the same, or look very much as

similar. So, as an evaluator, you don't have to have all of the repertoire and all of the characters and understand that. You are only comparing a finite set of strings. So, I want to clarify on that.

SARMAD HUSSAIN: Thank you, Edmon. Any more comments or questions?

MICHEL SUIGNARD: Yeah, well, you know, because like I did mention the Japanese case and just to make clear, I'm not proposing a solution here. I'm just saying that the Japanese GP should at least analyze it, and then provide their opinion, and if the opinion is to not create a variant relationship that's your prerogative. It's just that you would be – we want it to be covered. We basically want you to at least analyze the situation and come to your own conclusion and your conclusion will basically be submitted to the review, the public review, and then you should get reviewed one way or the other, and you would have to act on it.

That's the only thing we're asking, and those IPs are not really addressing these things. We're just, the only thing we're asking is that when we perceive this case where it could be perceived that there's a visual exchangeability, you have to basically document it. We believe that's a big thing and the conclusion is not forgone. The conclusion, I'm not trying to say like, you know, like at this

point, on the varied points. So, it can go one way or the other, but it has to be documented.

SARMAD HUSSAIN: Yes, please?

YOSHIRA YONEYA: This is Yoshira Yoneya. This is my quick point to Michel. So, in my opinion, the visual similarity between Hiragana and Katakana, by character by character, is not the matter, and the visual, right on the right, slide 11 or so, that kind of artificially-crafted story is a visual similarity, but it is not the realistic story because that kind of mixture never happened in the written or was used in Japanese.

So, this kind of application to the TLD is easily found, but I think that our analysis is that using Hiragana and Katakana [inaudible] in one label does not cause a visual similarity issue. But I'm not sure if this is a good explanation [of our] analysis, but otherwise I don't have any other explanation to explain Japanese analysis for the visual similarity. So, I'm really afraid of their using, just analyzing it, and explain to the public, but this understanding of Japanese and the understanding of non-Japanese is very different, so I think it is very difficult to analyze the work.

BILL: So, if I'm understanding what you're saying, it's that you can use, you can have a label with Kanji and Katakana, you can have a label with Kanji and Hiragana, but you can't have Katakana and Hiragana in the same label. Is that what you're saying?

YOSHIRA YONEYA: No, we, sometimes we use Kanji, Hiragana, Katakana simultaneously, but how we use Hiragana or Katakana is very different. So, this kind of mixture of the Kanji, Hiragana and the Katakana never happened, as in the original Japanese. So, I said this is a very artificially-crafted example.

MICHEL SUIGNARD: Again, that's the applicant's decision. I could have done the same slide, eliminating totally the first line, eliminating the right part of the second line, and you would have no idea what I'm talking about because you'd have only the visual this thing, and it would be applied by someone in the U.S., who knows where, that creates the label as a brand and the guy doesn't know better. And then he puts those three letters together and it's not Japanese, it's some guy that likes the name Taberu. And he saw that in some dictionary, only he made a mistake and he took the three letters, and he's not Japanese.

He's just – like, you know, it's like the ice-cream Häagen-Dazs in the U.S., it's totally meaningless and it's horrifying to all the Danish, and it still exists. This Häagen-Dazs is a totally brainless, meaningless sequence of letters. It means nothing, for sure. It's kind of pseudo-Danish or pseudo-Norwegian or Swedish, or whatever you want to think about, but it's totally meaningless. It is a brand that's totally known in the U.S., and you could expect someone to create something with it, even though it's totally meaningless.

This is the same thing. You know, I totally get that this is not Japanese. Don't get me wrong. I studied a bit of Japanese, enough to know that this is not meaningful. But, again think of just seeing the left part of the second line without knowing anything. You just saw that on a website, and there's nothing that's telling you that the bur is not a regular letter, but if you see it without knowing it, and if it's created by a non-Japanese brand owner, you have no idea.

So, I'm not trying to get a conclusion here. I want to be clear on this. You can have your own decision, and you can make it, and you document it, and then you have to live with what the public comment is going to be on it. Either people are going to say, yes, you're right, this is the right decision, you should go that way, or it would be the opposite. I'm not the one, the IP is the one that would decide either way.

YOSHIRA YONEYA: This is Yoshira, again. So, I think such kind of application it's almost zero chance to be happening. So, thinking about mostly their chance is overwork, I think, and such kind of mis- or meaningless applications should be, or have to be, rejected by the other partner because if the applicant did not know the wrong issue, exactly, they should not use that label, so that I think that it is not thought of in Asia, in the other places.

SARMAD HUSSAIN: We have Wil in the queue, then I am in the queue, and then Edmon.

WIL TAN: This is Wil Tan. I would just like to just point out that the LGR, this part of the work, what is in their scope is probably, it's more important for us to get the right rules in place, rather than looking at it from a very broad picture as to that these should never happen because our job is to create a secure and correct set of LGR according to the framework that has been set up for us in the procedure, and that's my comment.

SARMAD HUSSAIN: I'll pass on over to you Edmon.

EDMON CHUNG: Edmon, here. So, in response and actually as Yoneya said, there, for both IDN ccTLD and IDN gTLDs there's a requirement to explain what you mean by that string and explain how you're going to use it. So, those are the evaluation criteria for both new gTLDs and new IDN ccTLDs. So, I think, Yoneya, that point is important. But I actually put out my hand on a separate issue, since we're running, I think we are almost at time, right?

I just want to clarify one thing. Dennis reported earlier, there is a case where when someone applies for new gTLD, for example, and the table or the LGR is not ready, what's going to happen? I guess what you are trying to say is that if the LGR is not ready, the policy can decide whether to accept the application, but certainly not delegate it into the root zone, right? I mean, in order for it to be actually delegated in the root zone, I think there needs to be the Root Zone LGR. Am I correct? At least, for now, that's the suggestion.

DENNIS TANAKA: So, I don't have the exact – I mean, we, the last time we checked that point, I think we came up with three different scenarios, one of which is the one you are referring to, but I think as a group we haven't reached a consensus as to what one. I think the leading

thought is to present all three to the community to get their feedback.

MICHEL SUIGNARD: On that, by the way, we always when we do a review of LGR, we do make sure that the LGR passed the existing TLDs, or gTLDs. So, it's an obvious thing we do every time to make sure we're not making an existing delegated, or even applied-for, as much as possible, invalid by the LGR rules. So, we look at least as much as we can when the LGR is under process to make sure that we are not making something invalid because that would be kind of bad.

SARMAD HUSSAIN: Alright, so we are towards the end of the session, with a slow start, but an exciting finish. So, thank you all for joining, and we'll now close the session.

UNIDENTIFIED MALE: Reminder for ID program social which is when?

SARMAD HUSSAIN: There's an IDN program social at 6:30 tonight in –

[REGINA]: It's in Kairaku 1 in the main lobby of the Portopia Hotel and then you go down into the basement. It's there. The basement – this, yesterday?

[END OF TRANSCRIPTION]