JULIE BISLAND: Good morning, afternoon, evening, everyone. This is Julie Bisland for the recording. Welcome to the Latin Script Diacritics PDP call taking place on Wednesday, 08 April 2026. We do have apologies today from Louis Houle and Mark Datysgeld, and Sathees Babu will be joining late today. Statements of interest must be kept up to date. Does anyone have any updates to share? If so, please raise your hand or speak up now. Seeing none, if you need assistance updating your statement of interest, please email the GNSO Secretariat. All documentation and information can be found on the wiki space. Recordings will be posted shortly after the end of the call. Please remember to state your name before speaking for the recording. All chat sessions are being archived and as a reminder, participation in ICANN, including this session, is governed by the ICANN Expected Standards of Behavior, the ICANN Community Anti-Harassment Policy, and the ICANN Community Participant Code of Conduct. And with that, I will turn it back over to you, Michael. Thank you.

MICHAEL BAULAND: Thanks, Julie. Welcome everybody to meeting number 30. Next slide, please. The agenda for today is similar to the ones in the last two weeks. We will be recapping what we have been doing and then continue with a review of the public comments. Next slide, please. For last week, we talked about PR6 and the possibility to drop applications from the set, and we agreed to update the language delineating two scenarios. For IG8, our impact scenarios of evaluation outcomes were reviewed and we said ASCII gTLD is a requirement for the set to be intact because we

cannot have an LD set without an ASCII. That was a prerequisite of our charter. There was no need to apply EPP IDN's Final Draft 3.1 to the LD PDP because this does not apply to us. This was about the timing of the allocation. Next slide.

As action items, we are all asked to continuously review the public comment tool and the proposed draft. For PR6, as mentioned, we updated this due to a comment that we say this can be divided into one of the following scenarios. We used to have "into the following scenarios" to make it more clear that not both of them can exist at the same time. We were to examine the cases when ASCII or LD TLD already exists and the applicants apply for corresponding versions. This was mentioned by Tapani that this might not be covered clearly in the recommendations we currently have. We will see some suggestions in the following slides. Then we are examining options and provide paths forward for the scope of work regarding the limit or expansion of characters. Next slide, please.

Getting back to our PR6 recommendation, we had some changes, as you can see here with all the strikethrough and different colors and background. What we have right now is that PR6 says, "Consistent with Final Recommendation 325 of the PDP IDN Phase 1 Final Report, after submission of an application, the applicant is allowed to withdraw and apply for a gTLD string from that application, but is not allowed to add any other gTLD strings that were not originally applied for in that application. Only an applicant for .brand TLD string who has applied for ASCII gTLD string and all Latin diacritic gTLD strings are placed in a contention set is allowed to change its applied-for ASCII gTLD string and

its Latin diacritic strings under the conditions set out in SubPro PDP Recommendation 28." Next slide, please.

We then suggest to have these two scenarios not in the PR6, but make a new IG7 out of it to make it a bit more crowded in the PR. The suggestion is to have the wording which used to be in the PR6 now in the IG7, which reads: With respect to the withdrawal of an applied-for gTLD string per Preliminary Recommendation 6, it can be divided into one of the following scenarios. 7.1: If the application contains a single ASCII gTLD string and multiple Latin diacritic gTLD strings, and one applied-for Latin diacritic gTLD string is withdrawn from that application, the application will continue to be treated as a set under the exception process. That is the easy case. We have more than one LD TLD in the set, and then you can just remove one of those LD TLDs and the set still remains intact. 7.2: If the application contains a single ASCII gTLD string and a single LD TLD string, and either of the applied-for gTLD strings is withdrawn, the set will cease to exist and the application of the single gTLD string may proceed only through this standard process. This is the second case. If there is one ASCII and one LD TLD, then the applicant is allowed to withdraw either one of those and thereby losing the set and just having a single TLD. This is what we have been talking about in the last call and which we all seem to have agreed to.

Any questions here? Bill suggests adding an "or" after 7.1, just to be overwhelmingly clear. I am okay with that. You are the native speaker, so I trust that this will be okay. Someone put a suggestion in the chat. I think everybody is okay with the change here. Next slide, please. Now we come to the suggestion by Tapani that we should cover the cases where an existing TLD, either ASCII or Latin diacritic gTLD, already is in

the root zone or has been applied for in a previous round and now in one of the next rounds, someone wants to apply for corresponding TLDs to create a set.

For that, we have two suggestions here, two options. One is to add a few lines to this new IG7, which could read like: "With respect to applications for a corresponding ASCII gTLD or Latin diacritic gTLDs of their existing gTLDs, they will proceed in accordance with scenarios 7.1 and 7.2, except where an applied-for ASCII gTLD string is withdrawn from an application that also contains at least one LD gTLD and corresponds to an existing Latin diacritic gTLD. In such cases, the set would cease to exist and no application may proceed." This is Option 1. On the next slide, we will see Option 2. There we would not add this to IG7, but create a new implementation guidance for this case and this could read as follows: Implementation Guidance X, whatever that number will be.

After submission of an application for a corresponding ASCII gTLD or Latin diacritic gTLDs of their existing gTLDs pursuant to Preliminary Recommendations 3 and 4, the applicant is allowed to withdraw an applied-for gTLD string from that application, but is not allowed to add any other gTLD strings that were not originally applied for in that application. This can be divided into one of the following scenarios. 1: If the application that corresponds to an existing gTLD contains a single ASCII gTLD string or a single Latin diacritic gTLD string, and the only applied-for gTLD string is withdrawn, the application is no longer relevant. In this case, if you have an existing one and you want to add a second one to create a set, if you withdraw that one, then you just stay with the existing one and there is no application anymore.

2: If the application that corresponds to an existing ASCII gTLD contains multiple Latin diacritic gTLD strings and one applied-for Latin diacritic gTLD string is withdrawn from that application, the application will continue to be treated as a set under the exception process. Similarly, we have an existing ASCII gTLD and someone wants to create an LD set by adding more than one LD TLD, then they can, as in the PR6 scenario, drop any one of those LD TLDs and still continue with the application. 3: If the application that corresponds to an existing Latin diacritic gTLD contains a single ASCII gTLD string and at least one Latin diacritic gTLD string, and the applied-for ASCII gTLD string is withdrawn, the set will cease to exist and the application cannot proceed. If an applied-for Latin diacritic gTLD string is withdrawn instead from that application, the application will proceed in accordance with Scenario 2. Those are the two options. Any thoughts or questions? Anil, please go ahead.

ANIL JAIN: Thank you, Michael. On this particular page, I think whatever is written is acceptable to me. Can I see the previous slide? Here in the last, when it is written, "the set will cease to exist and no application may proceed." Can you please explain this particular part? This is not clear to me. Thank you.

MICHAEL BAULAND: Both should explain the same situation. Here we have this: "except where an applied-for ASCII gTLD string is withdrawn from an application that also contains at least one Latin diacritic gTLD. In such cases, the set

will cease to exist and no application may proceed." This is basically if you can forward to the next slide again, please. Thanks.

ANIL JAIN: I think it is already covered in the next slide.

MICHAEL BAULAND: I agree that the cases are a bit clearer in this version.

ANIL JAIN: True. I agree with you.

MICHAEL BAULAND: Bill, please.

BILL JOURIS: As I said in the chat, I think number two is clearer and Anil's comment merely reinforces that. If we are having challenges with number one, someone who does not have our in-depth experience dealing with this whole thing is probably going to have even more confusion. I would say go with number two. That seems like the way to go. Thank you.

MICHAEL BAULAND: Thanks. As Saewon put in the chat, we could also use the exact same wording which we have in the IG X now and put that into the IG7. If it is just the wording that is clearer in this version, then we could just use that in the other. Another question would be, do you think it should be

a separate implementation guidance, or should it be combined into one? I guess we all agree that the wording on this page is the way to go because it is more clear and it lists the cases exactly, but there would still be the option to use this wording as part of IG7. Any thoughts here or suggestions? Sarmad, please.

SARMAD HUSSAIN: Right. I am just trying to understand this last one in the next slide. We are saying that if there is an ASCII gTLD and the last one, can it also have multiple Latin gTLDs in that application, or only one Latin and one Latin diacritic and one ASCII?

MICHAEL BAULAND: No, it should be possible to have multiple LDs. This is a case where we have an existing LD TLD, and in order to create a set, we need at least a single ASCII because without it we cannot create the LD set, and at least one Latin diacritic gTLD string.

SARMAD HUSSAIN: I am thinking about a scenario where you have two Latin diacritic gTLDs and one ASCII gTLD, and the ASCII gTLD goes away, which is this scenario.

MICHAEL BAULAND: Which of those already exist and which are being applied for in your scenario?

SARMAD HUSSAIN: This one is actually suggesting that some of these exist. That is what I was missing. Thank you.

MICHAEL BAULAND: This whole part is about the question if a single TLD already exists and you want to create a set out of this existing one and one or more applied-for ones.

SARMAD HUSSAIN: Okay. Thank you.

MICHAEL BAULAND: You are welcome. Any other comments, questions, or are you in favor of having this in a separate IG, or would you like to have the text we are seeing now included into the IG7? Bill suggests to have this part moved into IG7. Thanks. Any agreement, disagreement, or don't cares? Tapani's and Bill's suggestions... Anil also suggesting including it like Bill suggests. Sarmad, please.

SARMAD HUSSAIN: I am not really sure. Maybe the policy team can advise better whether this should be a recommendation or an implementation guidance. This seems quite significant. Thank you.

MICHAEL BAULAND: Any comments by the policy team?

SAEWON LEE: I can speak to this. Again, I am so sorry, I was just looking for my hand and I could not. This is Saewon Lee from staff, for the record. Originally, we from staff had started from Preliminary Recommendation 6 and a great idea from Steve, actually, where he was because it was so specific and detailed and also trying to align with ICANN org's comment on being consistent with the EPDP IDNs language, we thought it might be better to have PR6 as the original recommendation that aligns with EPDP IDNs 3.25, and then separating the details of the two scenarios as well as this existing TLD corresponding guidance as a separate implementation guidance. That is the background of how this was divided. If we do decide to have a separate implementation guidance on this existing gTLD corresponding issue, maybe the best way to go is having a Preliminary Recommendation 6 and then having a separate IG7 that details all these issues so that we align with org's comment on being consistent with EPDP IDN's language while explaining explicitly the necessary steps. Does that make sense? Steve is here to explain further.

MICHAEL BAULAND: Steve, please.

STEVE CHAN: Thanks, Michael. Thanks, Saewon. This is Steve from staff, and really just to add to what Saewon's saying, the reason why we suggested to

frame it this way is sort of just a good practice for drafting recommendations. With the recommendation language, you generally are trying to capture the principle of what you are trying to achieve, which is the way that it is drafted now. When you get deeply into the specifics, that is generally where you want to consider making it implementation guidance because it is so specific. If you make it formal recommendation language, there is closer to zero wiggle room in case the working group did not get it exactly right. Once you talk about how to implement the recommendation, you will see in SubPro that this is the formula they took, where it is going to provide very little room to move around. They tended to make it implementation guidance, so that was part of the inspiration for why the staff team is suggesting this as an approach. Thanks.

MICHAEL BAULAND: Thanks, Steve. PR6 basically just says you are allowed to get rid of strings or labels, and the IGs are then explaining in detail how and what corner cases could exist if you withdraw TLDs under what circumstances this is possible. That is why the separation is there. Bill, please.

BILL JOURIS: This is Bill Jouris for the record. I understand what Steve is saying and that is exactly why I would like to have this as a recommendation separately because I do not want wiggle room. I want this done this way, period. Thank you.

MICHAEL BAULAND: Steve.

STEVE CHAN: Thanks Michael, and thanks Bill. I guess maybe to be a little more clear, it is not wiggle room that the org is looking for in implementation. I think maybe a better word to think of is rigidity, which can be a problem when you get into this level of specificity. The way that org approaches implementation is that they do everything in their power to implement as the recommendation and also the implementation guidance directs. But when you get this specific, there is a chance, which is why we are looking at this language again, that you miss things. This is the purpose of why the SubPro group provided certain things in the form of implementation guidance, because they recognized that they might not be infallible and miss a thing or two. The really important part that goes along with implementation is an Implementation Review Team, which is a community group that helps guide the staff team to make sure that they are implementing true intent. It is not that they want space to be able to stray from what the recommendation says. It provides flexibility in case that we, as a working group here, do not get everything 100% accurate. Hopefully, it helps a little bit with context. Thanks.

MICHAEL BAULAND: Thanks. Sarmad, Bill, are you okay with those explanations or do you have follow-up questions?

SARMAD HUSSAIN: I am good. Thank you.

MICHAEL BAULAND: Thanks. Bill is okay too, and I hear the majority is either indifferent whether we have a separate IG or added to IG7, so there was no one wanting to have this second part as a new IG. I think we can use the wording from the IG X and include that in the IG7. With that, we can finish this PR6 topic and go to the next slide. There we have the IG8, which we also looked at in detail last week. This is about the evaluation of the strings, and we added a new item here, 8.6, stating the evaluation outcome of an applied-for ASCII gTLD string impacts the evaluation of any other Latin diacritic gTLD strings in the application by causing the dissolution of the set. We saw all the possible examples last week where whenever you have one ASCII and two LD TLDs in a set, if one of the LD TLDs is not evaluated successfully, that is not a problem. This set can continue. The only problem occurred when there was one ASCII and at least two LD TLDs, and the ASCII was not evaluated correctly because then the set cannot continue and the set has to be dissolved and only a single LD TLD can continue under the assumption that it is evaluated successfully. Any questions or comments regarding this addition? Anil, please.

ANIL JAIN: Thank you, Michael. A small clarification. In the second example, suppose we have ASCII gTLD and two LD gTLDs which are applied for, and we are saying that ASCII gTLD could not go through because of whatever the reason is there. In that case, the set will get dissolved. Understood?

MICHAEL BAULAND: Yes.

ANIL JAIN: But in that situation, whether the two LD gTLDs can go together without a set or we have to process only one LD gTLD in this case. Just a clarification. Thank you.

MICHAEL BAULAND: Good question. Since we have the rule that we cannot switch the path between our LD exception process and the standard process, it is not possible to continue with two LD TLDs in a standard process because that would be switching the path. You would have to choose a single one and that would then be able to continue. But I agree that it is not totally clear here, and maybe we should add wording here, a footnote or something explaining this case. Because if you ask the question, most likely other people will come up with the same question and stumble over the same wording. Good point. We will adjust that and present you with a new version that takes care of that. Any other questions or comments here? I think we can continue. We anyway have to come back next week with a new wording that covers Anil's point, so we will take a look at this anyway next week again. Just in case you have some fundamental problems here, it is better to voice them now so we do not have to revisit that after next week again. Tapani agrees. I hear no objections, so we can continue to the next topic: the definition of diacritics. Next slide, please. We talked about this last week and the week before. It was also discussed in detail on the mailing list. Basically,

we now have two options to choose from because the third option here is just mentioned that in theory this might be possible too, and this is something that possibly other PDPs or other policy development processes could take a look at, but it is very clearly out of scope of our PDP because those characters mentioned there are not diacritics. They are just characters that also could map in some way to some ASCII characters, but that is not what we will be dealing with here. This option is just mentioned for completeness' sake and so it can be looked at in a different context, but not here. For our PDP, we basically have those two options here. The first option is to stay with the definition of diacritics which we have been using up to now, which means we look at the Unicode table and only if the character is decomposable into an ASCII character and one or more combining diacritical marks, or it is a combination of characters in the first instance which are not composed, then those are considered diacritics in our context and can be used for this exception process. The second suggestion brought up by Tapani was to expand the characters but with a limit, namely to also include those characters in addition to the ones we already have which are not decomposable but still have a name in Unicode which is of the pattern "small Latin character with [something]". An example would be small Latin letter o with stroke, this typical Northern European character. Before we take a decision here, some comments have been made in this context. Next slide, please. There is one by Christian. I am not sure if you want to talk about this yourself or if I should just read it.

CHRISTIAN DAWSON: I am happy to, but I think you could just read it. I tried to make it as clear as possible. Thanks.

MICHAEL BAULAND: Okay, thanks. Christian, as a council liaison, mentioned that there are some questions we have to consider when making this decision. One is: what constitutes a clear, objective, and machine-testable baseline? Second is: is the baseline sufficient for the working group to achieve its goals? And third: if not, what characteristics would the expansion of scope need to remain finite, predictable, and implementable? I see Bill's hand and then Christian.

BILL JOURIS: I am not quite clear what the problem is here. If we say it is decomposable or the description says "small Latin letter [whatever] with," that seems clear enough. Whether we decide to somehow parse the Unicode table names or whether we use those names and make ourselves the table, I do not really see either one of those as a particular impediment to making it machine-testable. I am not clear what the perceived problem is here. Thank you.

MICHAEL BAULAND: Okay, thanks. Maybe Christian wants to reply.

CHRISTIAN DAWSON: I am happy to. The only thing that I wanted to add in trying to explain the difference is I used the analogy in coming up with this of ensuring that in this process we can set what I referred to as a floor and a ceiling.

A floor would be standard due course common application, and then the ceiling would take into account the outliers. A machine-testable baseline could be something that identified standard processes and then was fed outliers, and that would be encompassing the floor and the ceiling. Ultimately, the process would be to take normal due course implementation objects as sort of element one and then outlier cases be laid upon it.

MICHAEL BAULAND: Thanks. Satish.

SATISH BABU: Thanks, Michael. Satish for the record. When I look at this, I sense a kind of a process problem rather than merely a technical problem of how we define these code points. For me, if this means... I am sympathetic in general to having a coherent, cogent kind of a definition that will serve us for not just now but maybe for the future as well, but subject to the constraint that the working group itself cannot enlarge our own remit.

I understand that this may not be an enlargement at all. My problem arises from the fact that we have decided on a definition. We have worked pretty much for a long time on it. We have put out the work to the community. The community has responded, but I do not recall a community response asking for this particular change. This is because we are all experts or linguists in our own right. We may have felt this is a need, but as At-Large, I am aware that the original impetus for this working group came from Canadian French speakers who wanted this

Quebec issue resolved. So there was a very real, palpable request from the community.

That is the basis of this narrow, very narrowly scoped PDP that we have right now. But here in Option 2, I sense that technologically it is not a closed issue; we can accommodate more technical features into this. But considering also the fact that there is a lot of anxiety about when we are going to finish our work, personally speaking, I have to balance these two as to what has been our remit. Are we in the process of redefining our own charter by going beyond? Technically, I am aware that this is strictly speaking not a redefinition. It is pretty much in line with what we ourselves decided in the beginning of the PDP. But then, having worked all this while, do we want to push our end of our PDP even further? We would have to then look at many of the implications of redefining what we started out with.

I sense a process problem here. First of all, it is not based on a powerful push from the community that we serve. Secondly, if it is going to make our work longer, given the fact that people are waiting for our work to be accomplished, we have to have a real solid reason why we take longer to complete this. Thank you.

MICHAEL BAULAND: Thanks. Good point. Before we start the discussion here for Options 1 and 2, I first wanted to introduce this comment from Christian, and there is also a comment by Sarmad on the next slide. Sorry Tapani, Bill, I will get back to you after we present the comment by Sarmad, unless you have a question here now and not a discussion point. If there is a

question to that slide, then please go ahead. If it is just discussing Options 1 and 2, then I would kindly ask you to wait until we finish this. Since your hands are still up, I assume these are questions. Then maybe quickly go back to the previous slide and give Tapani the floor for the question. Tapani, please.

TAPANI TARVAINEN: Looking at the questions here, I am a bit uncomfortable with saying "the baseline" here, but I understand the point being that you have a floor and a ceiling notion. Although in this case, the floor, the minimal set we could go with, is what we have: just the decomposable ones. But I would argue that my proposal is just as clear, objective, and machine-testable as well.

As for a ceiling, it could actually be taking outliers even beyond this, and I already agree that we should not go that far. My definition here could be perfectly finite, predictable, and implementable. Either of those criteria, decomposability and this "Latin letter with," are potentially changed if Unicode changes their definition. They might make a new character that is decomposable or they might add another "Latin letter with [something]." There is no difference there that I can see. In this case, both are equally finite, predictable, and implementable. As for the second slide you just showed where we would have to go with minimal... Can you show the next slide?

MICHAEL BAULAND: Before we pass the next slide, I would first like to give Sarmad the floor to present what he intended with the slide because that might already answer some questions. Sarmad, please take the floor.

SARMAD HUSSAIN: Thank you, Michael. I would like to take a couple of minutes to share a bit of background to contextualize this a bit. I am going to step back and talk about the development of Root Zone LGR, because this eventually also relates somewhat to the same exercise. Some of you have been involved in the development of Root Zone LGR. Root Zone LGR defines what are valid top-level domains and what are its variants. For the Root Zone LGR, there was a procedure developed called the Root Zone LGR Procedure, which defines how the Root Zone LGR should be designed. It follows some of the principles defined in RFC 6912.

I would like to highlight a couple of things which were part of the design of the Root Zone LGR for top-level domains. One principle was the inclusion principle. The conservatism principle generally works overall, especially for the top level or root zone. The inclusion principle basically discusses that there are two ways of managing characters in this set which can then create top-level domains. One is to start with everything and then exclude characters, versus the other design mechanism which is to start with an empty set and then only include characters which are absolutely needed.

Eventually, RFC 6912 and the LGR procedure are very clear that it has to be the inclusion principle. This means that for each script, you start with an empty set and then the community discusses and decides what are

the characters which are absolutely needed for the Root Zone LGR with evidence, and then includes those. If you go and look at the Root Zone LGR, it actually also documents the evidence for inclusion of each character, not just the character itself.

The second principle I want to bring up here was the principle of variants. The Root Zone LGR procedure basically says that as far as identification of variants is concerned, we should obviously maximize that set and block them. The reason is that wherever there is a chance of confusion, we should make sure that confusion gets addressed. If there are two different code points which are considered the same by the community, then they should get together in a variant set, and one of them should be allowed and the others should be blocked. The LGR procedure is very clear in saying that we want to maximize the blocked variants.

Almost like an exception process to that, it suggests that we could make some of those variants allocatable. It is very clear in the LGR procedure that the allocatable variants have to be minimized. Only those should be allowed which are absolutely necessary. Those are some of the design principles which are generally agreed by the community for designing how variant TLDs should be done.

In some ways, what we are doing here is creating an exception process to what is already happening in the Root Zone LGR to allow more TLDs to be applied for by the same applicant. This is an extension of allocatable variants. One way to think about it is that allocatable variants are an exception, and then this provides an exception to that

exception. We are getting into an area which really should be extremely conservative.

Of course, we need to make sure that the motivation is the underlying security and stability challenges. One thing which variants through Root Zone LGR address, which is not addressed by this process, is that in variants, the definition is very consistent. This means that if TLD 1 considers a character as a variant, that same character in TLD 2 will also have the exact same variant definition, meaning that variant sets are extremely predictable for end users.

In this particular case, what we are creating is an inconsistent extension where you can have a letter with a diacritic which one TLD applicant could say should be paired as a Latin diacritic with an ASCII. The same letter with the same diacritic in another TLD could be delegated to different entities. End users do not get a consistent experience across TLDs in using these characters. That is another reason why we need to be very conservative with this.

This is the first time we are doing it, and that was an argument for allocatable variants in the Root Zone LGR as well. This is the first time this is happening at the top-level domain name level. Top-level domain names have very significant implications because they can have millions of second-level registrations. We need to be extremely conservative to start with. Over time, when the community gains experience with this, it can be made more liberal.

Coming back to this particular point, using the same argumentation which we have had in designing Root Zone LGR, when we are

considering the set of diacritics which should be allowed, I am presenting a third option. This option is that we follow the same design principles as Root Zone LGR, which is an inclusion-based principle. We start with an empty set and only add diacritics as they are needed. At a policy level, the working group can create policy to guide how that set should be developed. Then a proper, separate community-based working group could do that analysis, like the generation panels did, to really define what should be allowed and why, documenting the reasons for each diacritic which is included. Let me stop there. It is a different viewpoint and is being presented for the working group's consideration. Thank you.

MICHAEL BAULAND: Thanks, Sarmad, for that background information and also the suggestion. Before I open the floor, just a quick question. Your suggestion would be that we, in this PDP, do not deal with the question of which characters or diacritics are in scope or not in scope. We just make the rules for some up-to-now undefined set and some other working group or generation panel would be responsible for defining which characters can be used in this policy we create. Is that a correct understanding?

SARMAD HUSSAIN: Yes, I think that is one option. The reason being that even Root Zone LGR is an incremental process. It can evolve over time, even Unicode evolves over time. If you make it part of the policy, then it gets stuck in the current Unicode version unless you have a mechanism to spin it off

into a separate working group from the community which looks and uses the same kind of principles and develops this. They can have one version at one point, and then when it is needed again, input triggered by the community, they can reconvene and add more diacritics as needed. One does not need to come back and update the policy every time.

MICHAEL BAULAND: Okay. Thanks for that explanation. With that, I think we can go back to Slide 12 with the options. We are missing the new option by Sarmad, which we can keep in mind. Back to the queue. Amadeu, please.

AMADEU ABRIL I ABRIL: I thought that Tapani and Bill were before me, but in any case, is it my turn? Thanks, Sarmad, but no thanks. I think that what you are saying is perfectly correct except that it misses completely the point of what we are doing here. Clearing with the Root Zone LGR, as far as I can tell, the Root Zone LGR for Latin does not decide that E with acute, small Latin letter E, and all the rest of the list we are discussing here had to be blocked variants. They decided they were not variants at all.

Whatever we do here is not expanding on what is allowed by the Root Zone LGR; quite the contrary. We are trying to put some restrictions just in case somebody applies for Latin diacritics together. We try to prevent the disperse that could be confusingly similar to the users but are not variants. We are trying to deal with something that the Root Zone LGR did not take into account. What we are doing here is a procedure for the exception to extreme similarity, not variants. When I hear some of

the comments here, it looks like we are trying to expand the base of available characters. No. What is not in the Root Zone LGR cannot be in a TLD. We are only dealing with things that are simultaneously allocatable, but we believe that some of them would be confusingly similar.

As we do not know what the string similarity panel could do, we suspect and are trying to say, "Look, we need to reduce confusion." We are trying to handle things like quebec and .quebec in a way that they are not confusing, not allowing them simply because the Root Zone LGR would allow them to coexist. I really do not understand what we are doing. We are not trying to expand. O with a stroke is there. We are trying to prevent O with a stroke and O from being used simultaneously for something that could be confusingly similar because it would be the same TLD at the end of the day.

In that sense, I understand your proposal, but I have to say, "No thanks." Just sending the results of this TLD to a lot of different TLDs... what? For O stroke, when we do that, then we go to the Faroe Islands, Norway, and Denmark and we ask them whether they can be confused or not. But the problem is not them. All the users would be using domain names with O stroke. It is not a question of the local communities; it is a question of the users. They would know why they need that.

Regarding this, I will not support having a table because having a table will be considered needed and does not have a method for amending errors that will always happen or expanding if something has been missing. We need criteria. The one we have today is there and is

workable, but probably still limiting. If the second one really does not pose technical problems, I would support the second one. We are not talking here about the second-level domains that are registered each day. We are talking about a one-time string similarity review to the TLD when it arrives. I think that the second proposal, which comes from Tapani with Mark's amendments and Michael's comments, is something that seems quite workable for this one-time test on the string similarity level, not the Root Zone acceptability of that character that has already been solved. Thanks.

MICHAEL BAULAND: Thanks, Amadeu. I guess Sarmad wants to reply to that. Bill, do you want to open a new comment or also reply to that?

BILL JOURIS: I have a number of comments, so I would say let Sarmad go first and then I can follow up.

MICHAEL BAULAND: Okay, great. Thanks for that. Sarmad, please.

SARMAD HUSSAIN: Right. I think one way to think about this and why this analogy would still work is that through Root Zone, what are we doing? We are determining what TLD can go into a root zone. With special considerations that in certain cases, two different strings, if they go into the root zone, first of all, there needs to be a very conservative

mechanism to allow that. In certain cases, if there are two different strings in the root zone, they must be managed by the same applicant or registry operator.

If you think about it from that perspective, I think what we are doing with the LD TLD is trying to do the same thing. We are trying to determine which additional strings, which are perhaps not allowed through the current process, could still go to the root zone and whether those strings need to be managed by the same applicant or not. In that sense, I think the Root Zone LGR work and this work are in some ways similar or parallel, where this particular process is allowing more strings into the root zone like the Root Zone LGR is doing as a baseline. Obviously, this one is more of an exception process to that. That is what I was suggesting. Thank you.

MICHAEL BAULAND: Sarmad, for the explanation. And then Bill, thanks again for waiting.

BILL JOURIS: Actually, I have three comments, so bear with me. First off, Satish expressed some concern about our extending how long our work will take. Thinking back over all of the stuff we have done, I do not see anywhere where changing to Option 2 would require us to go back and change anything else. If somebody has a specific issue, fine, but I cannot think of any.

Second, if I understood Sarmad correctly, what he is talking about would in fact delay the results of what we have got because what he

seems to be saying is, "Okay, you have to go back through all of the combinations and come up with a justification for each one." I just cannot see any reason to do that at this point. Finally, I would like to take what Amadeu said and, while there may be one or two nits that I would pick with it, basically I agree entirely. Thank you.

MICHAEL BAULAND: Thanks, Bill. No queue. Anyone else who wants to chime in and argue for... yeah, Tapani, please.

TAPANI TARVAINEN: As I am listening to Sarmad's case, I do not see anything in there that makes a significant difference between those two options, except just that Option 1 is a bit simpler. We have just one line. If we go to the floor and ceiling case, I would argue that the floor would actually be taking just the "Latin letter with" cases. That is a smaller set, and it is also a set of diacritics that includes the .quebec case.

As for the time it takes, as Bill said, I do not see anything that would change beyond making this one recommendation wording change. It would not delay us much at all. Maybe we would have to deal with the next meeting to agree on the wording, but it is not significant. The process issue mentioned earlier was that we discussed this earlier and I am reopening it as it were. As I said, I did not have time or nobody brought up some of the cases here. I was under the impression that Unicode actually defines diacritics as just the decomposable ones, which is not the case. That is why I felt it appropriate to reopen it. Otherwise, Bill has said everything I wanted to say. Thank you.

MICHAEL BAULAND: Thanks, Tapani. I think you are right in that sense that Sarmad's suggestion is not related to extending from Option 1 to Option 2, but it equally applies if we had not opened the discussion to include Option 2. It is also equally related to this Option 1 we have. It is not in particular to what you suggested. You are right in that assumption. Seeing and reading and hearing the comments right now, I think we have kind of two opinions here. The majority of the group members are in favor of Option 2. I have not seen any comment or any suggestion to stay with Option 1, and there are the suggestion and the caveats expressed by Sarmad, that this might be too much and should be limited to only what is necessary. Sarmad, maybe you can express this better than I.

SARMAD HUSSAIN: Yes, I would like to make two comments. We reached out to some Unicode experts on Option 2 just to check whether that would be machine processable, for example. One of the comments which we have received is that parsing letter names is not a predictable process because Unicode does not design letter names with the perspective that anything which follows "with" has to be a diacritic. It actually could be other items which may or may not be a diacritic. That was a perspective shared back from the Unicode expert whom we reached out to. I am just saying that in the context of going with Option 2, that is obviously an implication. Let me stop here and see if anybody has any comments on it.

MICHAEL BAULAND: Thanks, Sarmad. I see in the chat that Tapani asked whether she could see the exact wording of those experts, if you have that available and could send that to the mailing list.

SARMAD HUSSAIN: If I can find it in the chat.

MICHAEL BAULAND: Thank you very much. The other thing I would like to point out is that machine readability is not 100% required because this list we created does not necessarily have to be generated over and over again. We could come up with the list based on the current Unicode definitions and just keep that list as the one to go with as a fixed list and just have a rule with what to do if new characters get introduced into Unicode. But I see your point.

I also read Bill's comment: "If we define diacritics this way, do we really care if Unicode does something different?" Tapani writes: "As far as I know, the names of Unicode letters are pretty fixed. Very unlikely to change. Probably less likely than decomposability." That is an interesting point. I have no idea how this Unicode works, but in case the definitions of what is decomposable are more or less equally likely to change like the names, then this would indeed be an argument that makes sense. I also see Sarmad's comment regarding the exact wording. I will just read it quickly, and then I go to Amadeu.

It says here, "Not a good idea to data mine the names list for this. The way decomposition is made on the names is not regular; the concept of

diacritics may also be fuzzy. You may have a "with," but the following element is not a diacritic (example 00D8: LATIN CAPITAL LETTER O WITH STROKE). The description of the diacritics may also not be totally regular, thinking of the block 1E00 to 1EFF. The best way is to use UnicodeData.txt and Scripts.txt to determine Latin, and then look into UnicodeData.txt to see which Latin characters decompose and get the exact decomposition." Amadeu, we just have two minutes, but please go ahead.

AMADEU ABRIL I ABRIL: There's one short thing that I do not understand regarding the comment that Sarmad mentioned saying, "Oh, perhaps we add where you will find a small letter something plus something that is not a diacritic." The funny thing is that there is not a single univocal definition of diacritic in Unicode. There is a composition one, but not the diacritic word where we start with the problem. Second, if that is something bizarre, it will not work.

Take my beloved U+0140, which is the non-usable letter for the Catalan L-geminada, which is a partial representation, small letter L with middle dot. Indeed, the question is not whether it is a diacritic or not. The question is this cannot be used because it is outside RZ-LGR. So it does not matter. It cannot be used for our purpose because it cannot be accepted as a valid character in the Root Zone LGR, period. No matter whether it is a diacritic or not. In fact, it is not a diacritic; it is a punctuation sign, and that is the problem. I do not think that this question that something is not a diacritic is a real problem. The real question is whether that character would still be within the scope of the

original LGR. If it is and it is not blocked, it means it can be used for a TLD, period. Then we come to our concrete problem. Okay, thanks.

MICHAEL BAULAND: Thanks, Amadeu. We just have a few things left. Sarmad, if you can just respond in 10 or 15 seconds, otherwise we would have to postpone until next week.

SARMAD HUSSAIN: I will just be quick. Thank you. I wanted to get back to you saying that what I was saying would result in a smaller set than what these options propose. That may be true. But the motivation is not to create a smaller set. The motivation is to create a deliberate set, meaning that we only include those things which are actually needed. If the exercise may lead both options to eventually the same set, in any case, what I was proposing was an inclusion method, not necessarily a smaller set. Thank you.

MICHAEL BAULAND: Okay. Thanks, Sarmad, and sorry for overrunning by one minute, but it was a good and interesting discussion and we will have to continue the discussion next week. With that, I will leave you to your other tasks. See you all next week. Thanks for joining. You can stop the recording.

**[END OF TRANSCRIPTION]**