DEVAN REED: Good morning, good afternoon, and good evening. Welcome to the Latin Script Diacritics PDP Working Group call taking place on Wednesday, 30 July, 2025, at 1315 UTC.

We do have apologies from Juliana Harsianti, and Anil may be joining late. Statements of interest must be kept up to date. If anyone has any updates to share, please raise your hand or speak up now.

If you need assistance of updating the statements of interest, please email the GNSO Secretariat. All documentation and information can be found on the Latin Script Diacritics wiki space. Recordings will be posted shortly after the end of the call. Please remember to set your name before speaking for the transcript. And please note all chat sessions are being archived. As a reminder, participation in ICANN, including this session, is governed by the ICANN Expected Standards of Behavior and the ICANN Community Anti-Harassment Policy.

Thank you. And back over to Michael.

MICHAEL BAULAND: Thanks, Devan. Hello, everybody, to meeting number 14.

Next slide, please. So the agenda is quite similar to the last meetings’. We'll basically visit the TBD items from our chapter question three.

Next slide, please. That's the outcome we filled the spreadsheet [with], which is basically what we did the last meeting. And some key action items that have been completed are that the examples for the slides we had had been changed, adjusted to the comments provided. We had no meeting last week since I had the week off and was traveling. And that's also completed, obviously. And yeah, we are going to continue reviewing the TBD items.

Next slide, please. And with that, we can directly switch to the Excel sheet and take a look at the final Rec 3.4. That's where we left off. The discussion was ongoing and the question was whether there should be one application for all the strings that are intended to be combined via our LD policies or whether there should be a separate application for the base ASCII version and the Latin diacritic version.

And I already see Bill and Alan's hand. So please, Bill, kick off the discussion.

BILL JOURIS: Thanks. It has occurred to me that if we have one application, it will be a lot simpler as a matter of administration to keep track of the fact that there are multiple strings to be considered. If we have multiple applications, it's a little easier to lose track of “Oh wait, this is a variation”—yeah, variation; not a variant, but a variation—“on something else that was applied on a different one.”

So having them all together, I think, just administratively would make life a lot easier for whoever's considering these. Thank you.

MICHAEL BAULAND: Thanks, Bill. Yeah, that's a good point. It probably makes the process easier. Alan, please?

ALAN BARRETT: Thanks, Michael. I don't have strong opinions on this. I do think it may be administratively easier if there's one application, but I want to highlight the cost implications for IDNs. The IDN-PDP recommended that base application plus a small number of variants should be offered at a lower price than if they were completely independent. So this group may want to consider something similar for the ASCII diacritics. Again, I don't have opinions, but I think you should consider it.

MICHAEL BAULAND: Thanks. Yeah, the [inaudible] discussion is also one of the TBD items, but we will revisit that later. But of course it's somehow related to the application process, but we put that for later discussion.

So, yeah, Satish said that in the EPDP they also had a single application for the entire variant set. While that is true, the difficulty here or basically the thing we need to discuss and decide is also relating to predictability because if we say that they can have one application per TLD, then those applications could go through separately if they are not found to be confusingly similar in the string evaluation and just exist as two TLDs with no relation whatsoever at all, or they could be found similar and then the consequence would be that our PDP policies would come into effect, whereas if there were just a single application for the whole set, this would probably mean that the applicant would beforehand already state that they intend to run those TLDs as a bundle and work with the PDP policies created by us. And probably the string similarity checks for these two labels, whether they are too similar or not, would need to be done because we already say beforehand that we consider them similar and want to run them as a bundle.

Sarmad, please.

SARMAD HUSSAIN: Thank you, Michael. So, yes, I agree with what you're saying. But one other implication of that is that in such a case, one of the strings is primary, and other is, I guess, not a variant, but some variation or dependent string. And what that implies is that if the primary is, for some reason, undelegated, the other string, even though it's not a variant, would be undelegated as well, because that's, I guess, how our contracts probably also set up things.

MICHAEL BAULAND: Yeah, that's one point, although there could be policies that could take care of that, because unlike variants, we don't really need a primary because we don't have the disposition values blocked and allocatable, so there might be a possibility to deal with that with some different policies. But at least in the variant case, it's the case that we have a primary, and if that's undelegated, the whole set would need to be removed.

Edmon, please.

EDMON CHUNG: I think, Michael, you bring up a very good point in the sense that there are potential cases, at least, that a diacritic version is not considered a visually similar or confusingly similar string. In those cases, then, essentially, conceptually, they should be two completely different applications, and there could be competition for the other string, right? it is possible that that would happen. This issue seems to link to a number of things, including what Alan said in terms of cost, and including what Sarmad mentioned about the relationship.

Maybe we should think of another possibility where they go in as two different applications, but if they are found to be visually confusingly similar, then there's the mechanism to put it back together. And then we can think of a cost, whether we do a waiver or whatever, because if it is a completely separate thing, then there could be competition. And probably that shouldn't trigger a cost aspect because it wasn't in the first place automatically bundled together because it could be operated by two different registries.

So in essence, I think we need to think through this in tandem with a few other things, though I'm not sure we could single this out and just talk about this decision without also considering at least two or three of the other things.

MICHAEL BAULAND: Okay, yeah. Interesting points. Thanks. Bill?

BILL JOURIS: I think I'm agreeing with Edmon, although I'm not quite sure I completely understood his point. It seems to me if we put all the applications together on a single form, and then it's found not visually similar, there's no reason we can't make a policy that says, "Okay, we decided it's not similar. You have the option to put in a separate application. And tell everyone who's applying ahead of time that there may be updates saying, 'Okay, this one just became available. So you may be able to apply for that one as well.'" There's a lot of process running around, but I think we can manage to put something together that is not completely opaque. Thank you.

MICHAEL BAULAND: Thanks, Bill. Alan, please.

ALAN BARRETT: Thanks. I'm afraid I'm confused here. I thought that this working group was looking at ASCII diacritics (I don't want to call them variants), regardless of whether or not they're found to be very similar. So I thought that by definition, two strings, one pure ASCII and one with diacritics, would automatically be within the scope of this working group's recommendations, without the necessity to check for visual similarity. So I'm confused.

MICHAEL BAULAND: I think that's not necessarily the case because we try to create an exception process for the cases where two applications would be rejected because of being too similar to each other. So it's actually talking about the cases where you want to have two applications, want to have two TLDs, but they are found to be too similar, and we want to have an exception process that doesn't reject those or one of them, but makes it possible for the applicant to run both of them with some restrictions, which we have mostly already defined in the IDN EPDP phase 2 recommendations, which we have transformed to our case.

ALAN BARRETT: Okay, so I can certainly imagine situations where maybe two different diacritics are found to be not visually similar—I don't know, maybe a dot above a letter and a [tilde] above a letter, maybe not too similar. So, yes, it could happen. So then I just want to make sure that the output of the working group is very clear on what's in scope.

MICHAEL BAULAND: Thanks. Yeah, that makes sense.

Any other thoughts, opinions?

So, summarizing, it doesn't really seem we have a clear opinion here whether we want to have a single application or multiple. I personally, without my chair hat on, think that the single application has some benefits at least for the applicant and probably also for the process because then the applicant can already at the time of application know what kind of TLDs they will be running. If they use two separate applications, then they want to run them separately. And if they are too similar, then one of them gets rejected. But if they just have a single application, they know upfront that these two will be bundled, and this would also mean that they need an RFP that is capable of having the IDN level 3 support, namely to run variant TLDs. And if they would need to find that after the application, it might make their process more difficult.

But Sarmad, please.

SARMAD HUSSAIN: I'm just thinking whether if you do this, this remains in the scope. My understanding was that the scope was for those strings which actually go through the string similarity process and are found similar and therefore one of them cannot proceed. And so therefore there's all this exception process to allow it.

When we allow applications to be paired like this upfront before the string similarity process has happened, there's a possibility that two strings can actually be applied for which are not variants, through the same applicant, and maybe later on not even found similar by the Similarity Review Panel, which means that we are pairing strings which are not directly, at least based on my understanding, in scope of this PDP. Thanks.

MICHAEL BAULAND: Yeah, that's a good point, Sarmad. That might be the case, even though I think it's quite unlikely, but yeah, it’s definitely a possibility.

I'm just wondering if maybe the decision whether they are similar or not doesn't really need to be looked at in the case where you have a single application. Then you still do the strings similarity processes with all the other labels of all the other TLDs and applications, but you wouldn't need to compare those two TLDs to each other. So maybe that information wouldn't then even come out. But, basically, yeah, that's the thing that in that case we would have a policy for two TLDs which might not have been found too similar.

Satish?

SATISH BABU: Thanks, Michael. I'm trying to wrap my head around this. It's a fairly complex point as far as I'm concerned. I'd like to know what happens if two independent applications apply for diacritics of each other. Obviously they'll fail at the extreme similarity comparison level. But who gets to be the so-called primary? There's no primary here. But if one has to be rejected, then which one will get rejected? Thanks.

MICHAEL BAULAND: Okay, Sarmad, you want to respond to that?

SARMAD HUSSAIN: Not a response to that, but I just wanted to add that I think there should be still a mechanism which allows two different applicants to apply for two different strings, which have the same base characters but different only in diacritics and which are not found similar by the string similarity evaluation process. But in such cases, there are two paths. One is that such strings could be delegated very distinctly by two different applicants versus: if the same applicant applies for the same two strings in a single application, they can actually be bundled. So very different outcomes through the process.

MICHAEL BAULAND: Thanks. Back to Satish's question, I think it's not really defined which of those would be […] They would likely be in a contention set and then there would be the contention resolution similar to if you applied for two exact same labels. It would be a similar process, I guess.

So how do we continue here? I think I'm hearing both arguments, both for a single application with a problem of what if they are not confusingly similar and, with two applications, what if they are found similar, and how those two applications could then be combined into a single contract.

Edmon just wrote something in the chat. I'm reading this. "Just as a fictional illustration as I understand it, if Applicant A applies for dot-accent, Applicant B applies only for dot-accent, and Applicant C applies only for Accent from the LD version, if the two strings are not considered visually confusing, it's similar then it's possible that Applicant A will need to go into two different auctions against Applicants B and C separately."

Yeah, that's an interesting point. But even if they are found to be visually confusingly similar, Applicant A would need to be in an auction with B and C. I think the question whether they are confusingly similar or not doesn't make a difference for the outcome where A is in contention with both B and C, right?

EDMON CHUNG: Yes, you're right, but I'm just illustrating that it could be completely two different options and therefore it could end up with two different registry operators. And that relates a little bit to both when we think about the cost as well as whether [it's] two applications or also whether it's two different registry agreements, right?

MICHAEL BAULAND: Yeah, that's true. So if they are confusingly similar, then there will be a contention set of both three parties, but only one can win; either A, B or C. And if they are not considered visually confusingly similar, then there will still be a contention set. But in the case that B wins, it would mean that C automatically also wins, kind of, because A can't get their TLD—well, at least not both.

EDMON CHUNG: But if it's a separate application altogether, then A can win/beat C, but B can win/beat A. [It's] still possible, right? But if we go with Bill's idea, it's still possible that it comes in as one application, but when a situation happens like this, some mechanism needs to be there to split it into two.

MICHAEL BAULAND: Okay. Yeah. So if I understand you correctly, you suggest (and this was also suggested by Bill) to go with a single application process, but in case they are found not confusingly similar, Applicant A has the option to separate this single application into two applications? It's not mandatory because even if the TLDs are considered to be different, there's no rule. I think that this registry could run them the same way. I mean, that's also now possible. I think it was the case with this … OG? ONG, OGN? Something like that. Those were not confusingly similar, but they were run as a bundle, right?

[“][Company writes just a minor language in it]. Unsimilar is, let's say, not idiomatic there, but better would be dissimilar.[”] Okay.

So where are we? Going with a single application with the option to split, or two applications with the option to combine? Maybe we can have a quick show of hands for what your preference is. Maybe choose the vote button. Do we have this green and red yes and no button here? I'm not sure. If we do then—

BILL JOURIS: Michael, which is yes and which is no?

MICHAEL BAULAND: Yeah, I was just checking if that option is here, but yeah it is. So let's say the green yes is for a single application and the red no is for two applications. But first, maybe Edmon wants to say something and then we can …

EDMON CHUNG: Yeah, I did want to respond to Alan's note in the chat. I think that's precisely the issue that we're trying to solve because Alan, if we are going to go your approach, then essentially we probably need to update the LGR such that it would achieve the thing that you laid out, which is that it will be basically the same as IDN variants.

The problem we have right now is that because the Latin Script decided that they are not variants, then we have to have the visual similarity test because unless we do that, we don't know whether they can exist separately potentially for different applicants and registry operators because it is not disallowed by … If they're determined to be not visually similar, then nothing prohibits the two strings to be operated by completely different operators.

And I don't have a handy example, but I can totally imagine Vietnamese, for example (they have very elaborate accents), to have a string that is really not considered visually similar to a pure ASCII string. In those cases, then they could exist separately in the zone.

So I just want to highlight that.

MICHAEL BAULAND: Thanks, Edmon. Yeah, I agree with you. With the LGR, we definitely can't and don't want to change that and make these cases variants of each other, because then there would also be the question about allocatable variant or block variant, and we definitely don't want to go that way. But yeah, I basically agree with you here.

So yeah, let's just take a quick vote (it's not binding, but just the temperature of the room, so to say) on whether you think it should be a single application (then please tick your green box) or if you think it should be two separate applications (then choose the red box).

So, at the moment I've seen only green. Anybody in favor of separate applications? Please raise your hand or … Seeing none. So yeah, I think we have a solution here with the one application. [Good].

SAEWON LEE: Hi, Michael. Not that it matters anymore, but just for the record, I wanted to say that Asteway wanted the single application but with the possibility of splitting into two later on. I just wanted to leave that for the record.

MICHAEL BAULAND: Okay, thanks. Edmon, please.

EDMON CHUNG: And I just want to emphasize what Saewon just said, because that is important in order for me to think that a single application is good. I think we will need to think about a mechanism to split it into two if such things happen as indicated above. So the single author is predicated on the ability to split it later.

MICHAEL BAULAND: Okay, thanks. Yeah, we will note that. Sarmad, please.

SARMAD HUSSAIN: So I'm just trying to, I guess, see what is the criteria for including two different strings in a similar[/]same application. I guess what we're saying is the criteria is that the two strings have the same base characters, and they only differ in diacritics for Latin strings. Is that what we're saying?

Also, I guess the second comment, on splitting an application into two separate applications, I'm just thinking how easily that would be possible in an application round. But I guess that's a separate conversation. Thank you.

MICHAEL BAULAND: Good, thanks. Yeah. [Sarmad], on your first comment, that's right. It would be only possible for those cases where there's at least one base ASCII version of the TLD and then one or more diacritic versions where each letter is a diacritic version of one of the corresponding base ASCII version of that label. Sarmad, please.

SARMAD HUSSAIN: Okay, so if somebody is applying with a very clear intention up front that they want to run these different strings together in a similar kind of fashion like quote-unquote variants, then what would be the motivation to then split the application later?

MICHAEL BAULAND: Good question. I don't know. My personal preference was that if you say this upfront, you just stick with it. But I think Edmon made the point that if we don't allow this splitting, then we would allow labels to be combined, to be bundled, that are not confusingly similar, and that might not be something we want. But Edmon, please.

EDMON CHUNG: So in my mind, going back to my example of Applicant A, B, and C, the point of being able to split it is that Applicant A may end up only winning one option, and therefore it has to be split up or else it would force Applicant A to have to win both options for Applicant A to actually have anything at the end of the day. And that's the idea. That's the motivation for the ability to split it up.

MICHAEL BAULAND: Okay. Sarmad?

SARMAD HUSSAIN: Just again thinking aloud, couldn't what Edmon just said be achieved even without splitting the application; if the applicant wants two strings, which are, I guess, similar but not variants, they apply for them together, of course, with the intention that that's how they want to run it. In case it interferes or it is in contention with two different applications, that same string goes through an application process? And that is even true today. There can be indirect contention; that A is similar to B, and B is similar to C, where A and C are not similar and they all get bundled into a single contention set.

So in any case, even if there's a single contention set or two different contention sets where this application resides, they obviously resolve each of those contentions. And in case they lose one contention, [but when the two applications] are applied for together, one of them gets dropped, and the other applicant takes that particular string.

But in any case, I think the split of the application … I guess the reason I'm saying that is that, on starting with a single application and then splitting into two different applications, I'm not really sure how easily practical that is. Thanks.

MICHAEL BAULAND: Yeah, that might be legally or otherwise, process-wise, problematic. Edmon, please.

EDMON CHUNG: I think that's the reason why we need to think through this probably a little bit more as we talk about the cost and a few of the other things.

But I think one of the things that I think we need to avoid is [if] this is considered two non-visual similar or dissimilar strings [, where] Applicant B and C don't have to be forced into one contention set among all the three, right? So that is the key.

How we implement it, whether it's one, split in two, or however, I think doesn't really matter so much. Or maybe it is not so much a policy recommendation, but rather implementation guidelines that say, somehow, for the contingent set, even if they apply two strings which are diacritics of an ASCII string, there is a potential that they are forced to drop one and take the other by way of auction when something like this happens.

MICHAEL BAULAND: Thanks, Edmon. I'm just wondering, is it really a problem if that would lead to a single contention set, including B and C, even if B and C are not considered similar to each other? Because that can already happen now. If you have tables B and C, which are not similar to each other, there could be a label D, which is considered to be similar to B and considered to be similar to C. And in that case, also all three of them would be put into a contention set, even though B and C are not similar to each other. So there's nothing new we would create. If we say that, in the case, you created all three[, it] would just be a single contention set, and one of the three wins.

EDMON CHUNG: So yes, if there is a third string, right? But right now, my example is not the case. There are just two strings, right? So there's nothing creating that relationship.

MICHAEL BAULAND: That's true. There's no third string now. It would be just one combined application, which would cause a similar behavior as if there were a third string.

EDMON CHUNG: Then that is the new thing then.

MICHAEL BAULAND: Okay, yeah, that's right. I agree.

Tapani says we might have a case with a two-diacritics version, and only one of them is found to be similar to the base ASCII version. Not sure if it matters here. Yeah, good point, Tapani. This is of course, in theory, also possible, and we you might want to look at that too. I guess it makes sense to come up with some examples in the end and look at how they are dealt with in our rules, whether all of these are covered or if some additional policy or rule is missing to cover those.

So I think we just keep it now here with a single application, and we can still come back[,] whether we want the possibility to split it up once we have answered the other questions. Maybe that makes it easier then.

So the next point we have here is 3.5, where we say that in the IDN EPDP case, there are certain justifications the applicant has to give why as to they want to have a domain TLD together with a variant. They need to say the meaning or the intended meaning for non-dictionary words. For each of the [inaudible] tables, they need to explain how the primary and variant labels are considered the same. In our case, this might not be necessary because this could always be the reason ASCII is a fallback representation. And then there's the explanation of the benefits to the user communities and the description of the steps that the applicant would take to minimize the operational management complexities of the variant gTLDs.

So the question here is now, do we also want to require the applicants to justify why they want to have these bundled TLDs? Any thoughts?

The alternative if we don't use that recommendation in our case would be that the applicants can just apply for those TLDs and have no reasoning necessary and no checks for those reasons. Sarmad, please go ahead.

SARMAD HUSSAIN: So this particular recommendation for variants was based on a SSAC recommendation which said that having strings which are, I guess, "administratively put together," and where you have multiple variants at top level, multiple strings at top level, and then multiple strings at second level, create a lot of different domain names which are administratively in some ways bundled together. And that can potentially cause management challenges.

And therefore, the SSAC recommendation was that the number of domain names which are associated with each other—quote-unquote associated with each other—and managed by the same registrant, for example, should be minimized. And one mechanism to minimize that was to ask for clear reason. "These are some justifications which are needed for each in this particular variant string." And that justification allows to, I guess, not arbitrarily expand the number of variants which are delegated, but only take those variants which are actually needed.

In some ways, if you think about it in this case, it could be a similar situation where you actually have a base ASCII and then you can have an arbitrary number of diacritized versions, and somebody could actually apply for many of those. Then they actually could be potentially bundled, which means that if they're bundled at the top level and then eventually the second level, if that's what the policy decides. Then it creates a similar kind of problem which

SSAC was raising; that it creates too many domain labels which are administratively put together and become difficult for registers and registers to manage. And if it is leading to a similar situation, then some mechanism needs to be built in, like the one here for variants, to minimize the domains which have to be managed by registrants and registrars.

So in that case, based on that argument, something similar is needed. It could be a justification for each of the diacritized versions; for example, asking which community they're serving and, for example, which language and so on. But it could be a variation, of course, of this.

Just to summarize, basically what I'm saying is it seems like this current scenario would have the same underlying challenges, and therefore some mechanism is needed. Thank you.

MICHAEL BAULAND: Thanks so much for the background information of how this rec came to be. And hearing from your comments, it seems like we would need a similar one for our case to avoid running into similar problems with too many persons. Satish, please.

SATISH BABU: Thanks, Michael. So I, of course, completely agree with Sarmad. The reason why we had this justification clause was more as a measure of manageability of the variant set or the EPDP.

Now, my question here is, is it a real problem for us in diacritics? The kind of combinatorial explosion that we had in EPDP means the combination of different variant characters could lead to quite large variant sets, which would be impractical for a registry to maintain. And there we had a data-driven approach. We could check and find out how many; what are the numbers we're talking about.

So rather than taking arbitrary decision, would it be possible for us to get some idea of the numbers concerned? Thank you.

MICHAEL BAULAND: Thanks, Satish. I think it actually can be similar because if it's … Look at some ASCII character. Let’s say A. There are maybe, I don't know, four, five, six different diacritic versions of that. And then you could come up with a label, AAA. And if there were six diacritic versions, then each of the A's could be replaced by one of the diacritics. So there are three to the power of six, if I'm not mixing up my math here. And with longer labels and other characters, I think there's a similar problem of exponential numbers being possible. And that means we also need to have some mechanism, as Sarmad explained, to avoid somebody applying for all of these.

Tapani says there are even 20 different critics for A. So, yeah, you would have 20 to the power of three with just three-character labels. So it’s 8,000 possibilities for a three-character label. And with longer labels, it’s even more.

So yeah, the stream cipher in xkcd. But the stream cipher is not what we use as a background, but it's a good example. Yeah, I agree.

So it makes sense to have similar rules. We will come up with wording that matches our cases, but we will have something similar here.

So the next point is then the implementation guidance to that. This basically just says that the evaluation panel must include knowledgeable people to do the evaluation (or sensible) and to have a pass-fail scale for those questions. And applicants must pass each of the elements to be able to proceed. I think it makes sense that we follow the principle here. If we agree to have those questions, they should be handled in a similar way, I guess, unless there's some objection here.

Hearing none. So, yeah, I think, Saewon, you can put that[,] as a consequence of 3.5[,] we also have that for us.

So, the next point is 3.21. We have the 3.25 up there, but we'll come to that later. Let's look at the 3.21 now. “Only the protected organization on the list of strings ineligible for delegation are allowed to apply for the allocatable variant labels of their protected strings at the top level. Consistent with Final Recommendation 3.1, an application for an allocatable variant label of a protected string cannot precede an application for the protected string, which serves as a primary label for generating the variant label.”

Any ideas regarding this recommendation?

Hearing none. So I think we shouldn't make any exceptions here in our PDP for those processes, and we should just follow the existing rules regarding protected strings and have no exception process here. That would be my suggestion. For strings which are protected, they can't be applied for. And since we have no variant [label] relationship, there's nothing special here. Or Sarmad maybe comes up with an interesting point I didn't think of. Please go ahead.

SARMAD HUSSAIN: Taking the variant analogy to what we are talking about, this recommendation of course says that variants of protected strings are not allowed. So I guess a natural question in our context is , would a diacritized version of such Latin names be allowed?

MICHAEL BAULAND: That's a good point, but I don't think this actually is something we have to deal with in the PDP because the question already now exists. Are diacritized versions of [un]protected strings allowed right now? If they are, then we don't change it. And if they are not allowed, we also won't change it. I think this is not affecting our policy unless I'm misunderstanding this.

SARMAD HUSSAIN: No, no, I think you're not misunderstanding. I guess what I was saying was that in some ways we're expanding the scope of quote-unquote variants through this process. We are allowing other strings which are not variants to become associated with a set. And with that, could that cause confusion in such cases?

So for example, if there is a normal Latin string, and there's a same string with an accent, and those two strings are behaving very similar to each other, controlled by the same entity, and then there is this protected string, which obviously cannot be applied for, but it has this diacritized version which exists […] And then there is an expectation of, okay, so then maybe it is the same string as the non-diacritized version because of those other cases which now exist.

So I'm just thinking whether this creates scenarios which then need to also extend to some of these other restrictions. Thank you.

MICHAEL BAULAND: I understand your point. Thanks. And I'm also seeing Tapani's comment on whether there could be a protected string with diacritics where the base ASCII version is not protected. While all these cases could exist, I think there's nothing we need to do policy about because we just want to make an exception process in case two otherwise available strings are rejected because they are too similar to each other. So we don't want to extend the possibility to now make available a string which wouldn't be available due to current rules because of protected strings.

So anything related to protected strings would be caught even before our policy would come into effect. Either the existing policies would prohibit the strings to be allocated (then they're out of scope for us), or the current processes would allow them to be allocated. And in that sense, then we also don't care about the protected springs.

So unless I am seeing this incorrectly, all the questions regarding protected springs will already be covered by currently existing rules and out of some scope for us. It's different for the variant case because there we have a real connection between those where a protected string might have a variant. But this is not the case for us. We do the bundling after the process, after the check of the protected strings.

So my suggestion is that this is not applicable for us.

Alan writes, “I think that there will be objection processes that will catch an application for Red Cross, where Red Cross is a protected string. So no need to consider it.” Yeah, exactly. So Red Cross is a protected string. And if this diacritic version of Red Cross is applied for, independent of whether it's in a Latin diacritic bundle, there must be processes right now that either make that available or not, but we don't care for our policies here.

Seeing no objections. Edmon agrees here. Thanks.

So, 10 more minutes. I think we could come back to 3.25 now, which is between 3.4 and 3.5. “After submission of an application, the applicant is allowed to withdraw and apply for the variant label from the application, but is not allowed to add any other variant label that was not originally applied for at that application. Only an applicant for a brand TLD string who has applied for primary gTLD string as placed in a contention set is allowed to change its applied-for primary string and allocatable variant labels under the conditions set out in SubPro PDP Rec 28.”

So this is saying that with the variant case, usually you can't change the string you have applied for except if you are a brand.

And I guess this is not restricted to variants. But this is also the case for a single application. Please correct me, someone who is more knowledgeable in that process. But also [for] a single application where the string would be in contention, the brand would have the opportunity to change that.

Is that right? Can somebody confirm that or? Steve would probably know, but he's not here. Oh, Saewon says, "Correct." Thanks.

So, yeah, this basically says that brands have an exception to change their string. And the EPDP case said that, well, if they are allowed to change the string, of course, they are also allowed to change the corresponding variant labels because it makes no sense to have that mainstream change and not the variants.

The quick question is, do we want something similar in our case? If a brand applies for a base ASCII and the diacritic version, should they also be allowed to change that? Bill, please.

BILL JOURIS: I'm perhaps not understanding clearly what is going on here, but I can envision a case where somebody has the primary base Latin ASCII version and has applied for perhaps one diacritic version and then realizes, "Oh, wait, there's another diacritic version that I simply missed the first time through." And I don't see the problem with having them amend their application. This is not starting with the diacritic version. This is somebody who starts with the ASCII version and wishes to add an additional diacritic version.

So I would like to see us at least consider that that situation might arise. Thank you.

MICHAEL BAULAND: Thanks, Bill. Any other comments?

[A question to] Bill. You said you're not sure you understand what this is about. This is about the [fact] that brand TLD applicants have special allowances, a special rule, that in case their applied-for TLD comes into a contention set with another string, they are allowed to change their applied-for TLD to something which still must match their brand name, but which might be something different—maybe add something with a dash or whatever; if you have SAP, applied for it, and there's a contention that they might apply for software SAP instead; something like that. And this exception exists. And for the variant case, it was extended to include the variants.

And the question is, do we want to have the same exception process here and include the diacritic versions so as they can change also the whole set, so to say; what they applied for?

Yeah, it's getting late again, right? Yeah. Bill says he can also see something that is not a brand. But this special exception process from the SubPro PDP is just related to brand. So we don't want to extend that SubPro PDP rec to something which is not brand-related. This is a special brand thing, and therefore I think we should also keep with that, even if there are similar cases for non- brands. But since they are not applicable in the general case, I don't think we should make them applicable here.

But seeing that there are only three minutes left, I think we can come back to that next week. And with that, I hand over to Saewon. I remembered again.

SAEWON LEE: Thank you, Michael. Due to time limits, I won't go back to the slide because the next steps are actually quite limited today. Just please focus on reviewing the TBD items through this spreadsheet that you see on your screen. Those cells that are marked in red or orange are the ones that you should be focusing on. As you can see, the updates have been made with color changes as well, so please just focus on orange and red for the future.

And I will hand it over to John.

JOHN EMERY: Thank you so much, Saewon. Just wanted to say the major outcome today was based on the vote. Single application seemed to be the consensus. So action item as Saewon just mentioned is to please review the TBD items upcoming for our next meeting.

Michael, back to you.

MICHAEL BAULAND: Thanks, John and Saewon, for the summaries. And with that, I can end the meeting. And we'll see you next week. And I wish everybody a nice week and good weekend. Goodbye.

[END OF TRANSCRIPTION]