ICANN67 | Virtual Community Forum – UA: Technology and Email Address Internationalization Working Groups
Monday, March 09, 2020 – 10:45 to 12:15 CUN

UNIDENTIFIED MALE: Thank you, all, for joining this section on the UA Technology Working Group and UA E-Mail Address Internationalization Working Group meeting. I would like to hand over this session to Dr. Ajay Data, who's the Chair of the Universal Acceptance Steering Group. Dr. Data?

AJAY DATA: Good morning, good evening, good afternoon, everyone. This is a historic for all of us. ICANN is taking place remotely. It's an unprecedented moment in the whole world. The virus is infecting [our lives]. More than 100 countries are infected. By the time this was announced, India was not then. In country now, we have more than 40 cases and it's likewise everywhere. So we are in a very peculiar situation where the remotes-only meetings are happening, which is good for all of us.

Thank you to the Board for thinking it through. We must thank CANN Org, who organized this meeting in a very short span of time in a completely different way altogether. I am told that it is one of the largest online meetings ever which has happened in the world. So we are having thousands of people now having a meeting online. I'm sure [this] is going to follow in this.

I'm aware that ICANN org is working day and night to have this virtual meeting and have it as smooth and productive as possible. I'm also

Note: The following is the output resulting from transcribing an audio file into a word/text document. Although the transcription is largely accurate, in some cases may be incomplete or inaccurate due to inaudible passages and grammatical corrections. It is posted as an aid to the original audio file, but should not be treated as an authoritative record.

really thankful for everybody for joining at very difficult time for a lot of people, [not a] convenient [day] when we are physically meeting face-to-face. This is an unprecedented time we are having.

Before we get into the real discussion today on the Technology Working Group and UA, I will take a minute or two to explain for people who are joining for the first time what UA is and how they can participate.

Universal acceptance is a concept where all the domain names and e-mail addresses are treated equally. We have a website called UASG.tech. You may like to visit that website and read a lot of content, a lot of literature, a lot of knowledge with articles. A lot of history is there.

Dennis has just put in the UASG.tech URL link in the chat., if you want to click and visit that website. Dennis is the Vice-Chair of UASG. Thank you, Dennis.

This is teamwork of volunteers. I am also one of the volunteers. We are heading the chair: myself and two more Vice-Chairs. We have Mark, Dusan, and Dennis. We are all doing the admin stuff and are supported by almost 500 people in the list we have in the volunteering work. So this is one of the largest communities we are part of. I'm sure that we all are going to take this initiative of UA, which is not only going to break the [language barrier] but also bring billions of people online. We have a case study for that which shows a large opportunity for becoming UA-ready and taking these initiatives in your region, in your country, to take it further.

It is also very important to know that this started around four years before, in 2015. We are now five years now, almost. We have an annual strategic action plan. If you would like to go on the website and see the plan, you will know what are the focus areas, what working groups we have, and what is an interesting area for you. You are welcome to join those working groups. You're going to see, if it's the first time on your call, that the Technology Group and the EAI Working Group is going to present their update and the developments that have happened in the working group. If it interests you, then we are more than happy that you join that working group. You are welcome to join that working group.

With these thoughts, I would like to hand it over to the Technology Working Group Chair. They can run through it. I'm sure Sarmad Hussain, who is the Program Director and also heading the UA as an admin person from ICANN, is going to [attend to you for this portion.]. Thank you very much.

SATISH BABU: Hi. This is Satish Babu, the Chair of the UA Tech Working Group. I hope you can hear me.

AJAY DATA: Yes, Satish.

SATISH BABU: Thanks, Ajay. First of all, a warm welcome to everybody who has joined us today. We had a very active public forum just now [inaudible]. I see some of our ICANN community leaders also here in the session with us. I'd like to provide you with an update over the next 14 to 15 minutes about the work of the Tech Working Group, which is one of the six working groups with the large UASG community.

Here's the overview of what we're going to present: the context that we're working in and the charter, the overview of the pieces of work that we were supposed to take up, the current progress, and what we're doing jointly with the other working group within the UA. We will also have maybe some time for questions and answers.

Yes, please. Next. This slide would be very familiar to those of this in the UASG Working Group because this sets the context for what we're trying to achieve. If you look at this graph, it basically says what percentage of websites are actually accepting the different versions of internationalized e-mail addresses. You have different categories of the types of addresses. You have ASCII@ASCII.newshort. You can see the legend below.

From this graph, the immediate takeaway is that, when it comes to the Unicode-based names, it's less than 15% of the sites that are able to handle these e-mail addies. That means we have a huge gap here and we need to work to make this much, much higher so that the internationalized e-mail addies work just like any kind of e-mail addies. So, basically, this slide sets our context.

Next. What did we set out to do? I think this group was formally launched sometime in the fourth quarter of 2019, so we have not completed our first year. But, of course, some of the work that we are doing or planning to do has already been started much earlier, as Dr. Data said. The work actually started much earlier. We are actually continuing some of those pieces of work.

Here are main outcomes that we wanted to have. The first is to develop the scope of UA readiness and its testing criteria. This is jointly with other working groups. Now, the UA readiness is actually the heart of the activity of all the working groups. Even for us, it's a very essential part. Now, this is actually a moving target. It is in some kind of perpetual [inaudible]. You can't say it's complete, but it's ongoing. It's evolving. Then, evaluate and address technical standard and best practices. Then, evaluate and address the following open source and the proprietary applications on mobile and computer platforms.

So, basically, the idea here is that we try to look at the scope of UA readiness, and then we look at the standards and best practices and also where there is a major shortfall in terms of the popular software that we use. It could be anything from e-mail systems and webmail systems. If you look at the top-ranked websites, how many of them are able to handle … And which are the languages and libraries and frameworks that need addressing? Because, in terms of impact, there is actually some amount of divergence. There are some libraries and languages that, if you are able to remedy them first, the impact of those will be magnified because they are very popular. So that prioritization is part of our work.

In the last several months, we've been actually doing most of this work: basically looking at the application development environments, languages, libraries, and frameworks and also some of the applications, including fundamentally browsers, social media, content management systems, databases, and so on.

Of this, browsers, for instance, is very critical category because everybody uses browser and most of the work today is done on browsers. We have sometimes issues, like … One of the things that we were discussing recently was that the HTML5 standard has an internal e-mail validation mechanism. So, rather than manually let JavaScript validate it, the standard provides for e-mail validation.

Unfortunately, that validation does not cover the internationalized e-mail addies. So we need to work with the major standard organizations who are actually controlling the development or rather the implementation of these specifications to get the internationalized e-mail addies also acceptable there.

Then, develop technical documentation and develop technical training material. In this presentation, after my presentation, we'll be talking about the first of the training materials that we have developed.

Next. We are guided by our FY'20 action plan, and we are working towards whatever was actually stated in the workplan. We are trying to cover those. You can see that the ones that are in color are the ones that are in progress. The two black items we actually haven't got started with yet. Let me quickly run through this. The UA readiness, the first one, is actually joint work with other working groups. The second is to review and revise [inaudible] on UA readiness definition which case be used by developers. We haven't got started with this basically because the other ones are high priority. Some of them we could get started with and then farm it out to external contractors and then come back to these. So, in that prioritization, these two have actually fallen behind. Once we complete the outsourced or contracted work, we'll come back to these.

The third is to create a UA-readiness maturity model. This is also something that has come up. Dr. Data has actually informed us of the need for some kind of [inaudible], etc., which will come under this maturity model. So we are actually looking into this and we will be taking this up very shortly.

Identify [the eleven] standards and standards bodies [inaudible] updating these standards. Now, in terms of remediation, on some things, we can directly take up the work and contract it. On other things, we do not have the ability or the capacity to directly make these changes ourselves. We have to persuade some of the standards bodies and work with them. That is actually quite challenging. We are currently in the process of shortlisting which are the important standards bodies with whom we need to work and get their consent, support, and cooperation in making these changes for us. So that is ongoing now.

Review and remedy popular open source programming language, libraries, and [inaudible]. We will see in the next slide the work that is done already. Develop technology training material for UA readiness. Also ongoing. Develop communication material to reach UA Technology Working Group stakeholders in collaboration with the Comms Working Group.

Now, one of the major tasks that we have to get done is basically to reach out to the community of programmers. Earlier, we were looking at various ways of contacting programmers, maybe through tradeshows. Now, with the coronavirus, I don't think there's going to be any more large meetings or gatherings. The other was to work with universities, etc.

This communication is actually quite key. We are working jointly with the Comms Working Group, with is a very specialized working group for outreach, to make different kinds of messages in different kinds of media and delivery modes so that they can reach out to the appropriate group of people that we are targeting.

Next. We're trying to capture the current progress right now so far this year. First we started off with working on our charter. That took us several weeks or, I think, a couple of months. This working group meets once every fortnight. Those of you who would like to join this group, please go to the UASG.tech site. Scroll right to the bottom. You'll find a link that says Join Working Groups. Click on that and choose the working groups that you want to join. Then you can get added to the mailing list and you'll get invitations for the meeting. So we started off with the charter.

Then we had finished one piece of work, which is currently waiting to be assigned to a contractor. This is the programming languages and the frameworks. Even before we got started, Java [(or JS)] and Python had already been sunrise. So we have taken up a subset of the more popular programming languages and frameworks. Now, obviously, this is a very subjective and potentially controversial list of topics. So we had several rounds of discussion in our calls. This basically reflects the community consensus on what this group … And I think this group, most of us, are actually people with some technical background. Of course, we didn't have perfect agreement among ourselves on what are the most popular language. JavaScript comes in multiple libraries. Here we have provided for [inaudible] but Node is a notable omission. I know many people who are big fans of Node or JS, but we have not taken it because that'll be taking on too many things. So, in that sense, this is a reductionist kind of list: Java and Kotlin on Android, Swift and Vapor, C#, VB, and ASP, and .NET core, Go and Gin, PHP, and the mailing libraries for relevant platforms.

So this has been our first output. Currently, we are discussing the organizations and the standards bodies. Hopefully, by the next month, that statement of work will also be published. You can see these statements of work on our website.

Next. The training on universal acceptance of Java developers. That is going to be presented today after my presentation, so you can take a look at that. This is targeting Java developers and people who [inaudible] [projects] in Java. What]could be covered] … These are the kinds of things. So, basically, the idea is they should be able to handle in their code using the recommended practices: the internationalized domain names as well as e-mail addies. We have these five words which are there on the UASG site. So those are the actions that must be done, actually. So the code has to ensure that those five words are taken care of. The goal, of course, is to develop UA-compliant Java applications. By mid-April 2020 [it's going to be completely done].

Next. In the structure of the UASG, apart from the UASG leadership, which is the Chair and the Vice-Chairs, you also have a coordination group, which consists of the Chairs and Vice-Chairs, plus the leadership of all the different working groups. Many of the key cross-cutting concerns are actually discussed in that coordination groups. We also work hand-in-hand with the other working groups, particularly with the Measurement Working Group. Like I mentioned, we are working with the Comms Working Group on finetuning the messages for outreach.

Next. So that's it. If you download the presentation, you can click on the link to join us. I'd encourage anyone with the aptitude or inclination for technical stuff to please come and join us. We would like to have your opinions and your inputs on our work.

Sarmad, I don't know how much time we have for questions. Do we have any time?

SARMAD HUSSAIN: We have one minute.

SATISH BABU: Okay. Are there any questions?

PITINAN KOOARMONPATANA: We have one question.

SATISH BABU: Please go ahead.

PITINAN KOOARMONPATANA: This question is from [Vanda Carracayan]. The question: "I didn't know notice the example of ASCII@IDN.IDN on the slide." And she gave some examples.

SATISH BABU: Can you go back to that slide, please, Sarmad? So you're saying—oh, okay; next; yeah—ASCII@ …

PITINAN KOOARMONPATANA: ASCII@IDN.IDN.

SATISH BABU: Okay. Yeah, that is not there. But I think that is almost a trivial case because @IDN is covered in Unicode. ASCII is a [substantive] Unicode in that sense. So that should be covered in the last but one Unicode@IDN.IDN bucket.

SARMAD HUSSAN: Actually, the closest one would be ASCII@IDN.ASCII.

SATISH BABU: Oh, that's the third one.

SARMAD HUSSAIN: Because the requested e-mail was ASCII@IDN.IDN. So I think the domain names IDN and the mailbox name is ASCII. So the third one would be the closest [inaudible].

SATISH BABU: There's a question: "Why would that be the closest?" Sarmad, maybe you could answer it.

SATISH BABU: As I just said, that mailbox is ASCII, and the domain part is IDN. So the third one maps closest to it.

SATISH BABU: Yes. My first instinctual answer that I had was it was the last but one because the IDN.IDN parts are the same. What is asked for is actually IDN.IDN to the right of the @. On the left, instead of Unicode, it is ASCII.

But, anyway, I think we don't have time to have a more in-depth discussion on this. But, if we have time at the end of this presentation, we can take this up and come back.

Okay, Sarmad. Back to you.

SARMAD HUSSAIN: We actually have a hand up.

PITINAN KOOARMONPATANA: Yeah. We have Mark in the queue. So, Mark Svancarek, go ahead, please.

MARK SVANCAREK: I just wanted to mention that the ASCII@IDN.IDN or Unicode@IDN.IDN is really a terminology issue, for the most part. We'll talk about it in the EAI discussion. Thanks.

PITINAN KOOARMONPATANA: Thank you. Then we have another question in the chat from Maria Kolesnikova. "When we talk about UA development, we talk about migration from ASCII and Unicode in e-mail and domain name in general. Is that right? If so, what challenges may be faced after UA readiness will be reached technically when business and end users start to use EAI and IDNs widely and globally? Do we have all essential policies and procedures for that?" End of question.

SATISH BABU: Thanks for the question. I'm afraid we have a big bunch of speakers waiting after me. But we will park this question, and I request Sarmad to please respond once it is the end of this presentation.

So back to you, Sarmad.

SARMAD HUSSAIN: Okay. Over to Marc.

MARC BLANCHET: Thank you, Sarmad. Can you confirm you're hearing me?

SARMAD HUSSAIN: Yes.

MARC BLANCHET: Thanks. Next slide, please. This is a short presentation on training for Java developers for UA. The goal of the work is to create training material that is targeted to Java developers for properly handling the universal acceptance in their software while they develop their software. It's a bit more than 100 slides and some code examples. The common libraries used in Java are discussed so that people that are using some of the libraries they're used to will be able to assess if they are enough or ready or not. And some discussion on best current practices.

Next slide. The tutorial was sent to the UASG various mailing lists for, I think, a couple of months now. Thank you very much to everybody who reviewed it. The tutorial is being updated based on the reviews.

The overall table of contents or the plan of the tutorial is as follows. There is an initial section that describes the key format and the concepts related to universal acceptance. So it touches on Unicode, IDN, and EAI. Obviously that section is kind of dangerous because it's oversimplifying many stuff. But the idea here was that, for many Java developers, they may not be aware of the [inaudible] of this. For example, Unicode is two- or three-slide [inaudible]. So it doesn't go into the details, which is oversimplifying, but actually gives them a good primer of stuff that are relevant and important to know. An example of these are Unicode, UTF-8, and [inaudible], obviously with very basic details.

Then it goes into validating UA identifiers as input. So it does describe the issues of validation of UA identifiers [inaudible] about UA identifiers essentially are either domain names or e-mail addresses that are UA. Then it also discusses using the identifiers—for example, resolving domain names or sending e-mail.

Essentially the final section is about best practices. It actually applies to Java but essentially could apply for any programming language. Then there's conclusion and references.

Next slide, please. Here's some of the libraries discussed. This is not a full list. People developing Java application are well aware and know those kinds of libraries—the base IDN library—Apache Commons, [inaudible], ICU, Guava from Google, the native Java URL and URI libraries, some HTTP libraries, the mail libraries, and the Spring framework—Spring being very well used in the Java environment for essentially backend stuff. And there's a specific discussion about usage of Regex, which is another big issue where developers tend to use a basic Regex based on their knowledge of what a domain name and e-mail address is. In fact, doing a Regex for those kinds of cases, if you account for the UAs, is very difficult or too simple. So there's some words about those and examples of many Regex that are used in the field that are not compliant. It explains and shows why.

Next slide. So I think that's it for this Java tutorial. Again, the tutorial was sent to the various UASG mailing lists, so please review. We're finishing it at the moment, so it's almost time if you have additional comments.

Back to you, Sarmad.

SARMAD HUSSAIN: Thank you. We have maybe time for one question, and then we can take the rest after the presentations.

PITINAN KOOARMONPATANA: Sure. We have one question from [Harish] Choudhury. "How are we handling Unicode Latin [IPA extension.aresenseevooday." End of question.

MARC BLANCHET: I think that question is not relevant to the Java tutorial. I don't know who should be the right person to discuss that. I think that's a more broad question.

SARMAD HUSSAIN: Okay. Thank you, Marc. [With a request, you do] continue and explain the work being done on UA readiness conformance scoping.

MARC BLANCHET: Thank you. This work was essentially triggered by—that's my words—trying to … While the community was discussing UA readiness inside applications, how should we frame the discussion and also the testing and reviews and stuff? So the goal was then to define a framework where it defines the boundaries to verify UA readiness of any software. That way, we can apply that framework to our work in various discussions on software which are either native or web or as a database or user interface—all those kinds of things [they document as [inaudible]].

Next slide, please. The structure of the document as a short section on what the UA means so that it sets the stage of the discussion. It defines essentially two types of application. The early versions of the document actually defines more than two, but we finally aggregated it to two, which is essentially web-based. An application is either for the purpose for the discussion of that framework. An application, either a web-based application or native … In native, we include the mobile environment. Obviously, as the document [inaudible] there is a [blue frontier] between the two because there is, for example, applications that have web browsers embedded in them. Those kinds of variations are discussed in the document.

The next section describes the various components for each type of application. Obviously, it's a high-level set of components. Sometimes the components are glued together where you cannot really cut them in between smaller components. So it all depends on the actual application. It defines steps for processing UA identifiers. Those steps were defined by the community before. The only thing we added is another step, which is a process between the store and display to make sure we are gating each step towards the whole spectrum of processing steps within the application.

After defining steps, we identify the gates for testing the UA readiness to each application. Then, for each gate, we list a test case. A word about the test case here. It's more a high-level few test cases. It doesn't describe the detailed test cases [or] giving it direction of what you should accomplish at the gate in terms of the kind of test case you should apply. But it's obviously not comprehensive of test cases.

Next slide. Going a little bit deeper into the content, the web-based application has the following components. We define the following components, one of which is the web browser. Obviously may have UA readiness issues. The front end—again, that's then typical way people will define their applications for web applications. But, obviously, it's oversimplified. It just gives a good set of components. The front end is the component which creates and enables code to the executed in the browser. So it's responsible for creating HTML, CSS, and JavaScript code executed in the browser. The backend is the component running on a remote server, which typically executes more complex tasks—anything [inaudible] and have a database and stuff.

The document goes into some of the variations of this. For example, we talk about the fact that the web browser may have local storage. Therefore, it may be relevant to look at this if UA identifiers are actually stored in local storage. Those kinds of thing.

Next slide, please. The web-based application could contain, for example, a database. It could have access to the file system. An example of those two would be that the database may not be configured for Unicode or UTF-8. There might be some fields that are ASCII. Therefore, it you put UA components or identifiers in the database using ASCII [rows], then it will not work. You will lose the UA capability – file [inaudible] similarly, if file names are used and the file system is not UA [inaudible] ready.

One important component which is obviously more difficult to test, if you do open source testing, is an external service. Nowadays, there's a lot of web applications that are using various kinds of external services. They could have impact on UA readiness of the application. A very good example of this is the identification services, where, instead of doing your own [inaudible] authentication, you use other providers—authentication providers such as the big online companies (Facebook, Google, or private or enterprise kinds of authentication).

Therefore, you [pass] identifiers and they may not support EAI identifiers, for example.

Next slide. Native application components. Obviously this one is a little bit more difficult to cut in pieces. Essentially, you have the user interface, which collects then user input, the internal, which processes the user input and calls the database or stores in the database—the database as before.

Next slide. File system and external services. So it's similar.

Next slide. Here's the visual approach of the gating. So you have the various steps and then you apply various tests at then different gates or steps: accept, validate. You accept the user interface, what user types. So, if that person types "UA identifier," then you [make sure that] it is well accepted. An example of Satish said before was HTML. The e-mail form input is not UA ready. That's an example. Then you validate the user input. That's some example of where people use Regex. That sometimes doesn't work. You could process the different identifiers, store the database, and process back for display at the end.

Next slide. I think that's the last—yeah. I'm seeing a lot of chatting. Obviously I haven't followed it. Sarmad, back to you on questions or …

SARMAD HUSSAIN: I think we'll take questions toward the end to make sure that we get through all the materials. This is the, I think, last one on the technology side. We have Dean online, who's going to be presenting on the ICANN UA readiness update. Dean?

DEAN EDWARDS: Hi. Just confirming you can hear me.

SARMAD HUSSAIN: Yes, we can.

DEAN EDWARDS: Thanks, Sarmad. Hi, everybody. Dean Edwards here. For those who don't know me, I'm Senior Product Manager at ICANN. I am focused on this, amongst some other things. I'm reporting into Gary Petzer, who you would have heard from at the last ICANN meeting as far as our update goes.

Next slide, please. Today I'm going to quickly go through just an overview of our strategic focus from an ICANN perspective, the readiness update itself of where we stand as well as our internal systems and our public-facing systems, and then an update on our EAI readiness.

Next slide. I think a lot of you are already aware of this, but this just reinforces here ICANN's commitment to universal acceptance and to just really show you that it is a strategic focus for ICANN. As part of that, obviously we are ensuring that all of our custom systems are universal-acceptance-ready. We are going A-Z, through all of those, and making sure the ones that are applicable are ready and available.

We also are making sure that, as you might accept, universal-acceptance readiness is part of our guidelines and is built in as we design and develop any new systems or anything that's coming up through the pipeline.

The other thing we're doing that we've had in place now for a little while is part of our procurement process, when we work with outside vendors, where we have the universal-acceptance readiness provision built into that. So, when we're talking with any vendor, we make sure they understand we have a requirement for that support as part of our development and their commitment to us.

Then we briefly talked about custom systems. We're also working externally with any vendors we do have working with us where we use their systems that they understand what universal acceptance, they're ready for it, and, if they don't know what is or they need some education, we are working with them to make sure they're up to speed. Really, the focus is there getting, if they don't have it, universal acceptance on their roadmaps. So there is an effort to do that within the team.

Next slide. As far as our systems specifically at ICANN, on our approach, we've gone through a couple iterations of this. This is part of our learning exercise here for us. We have split our readiness approach to several phases. These might be obvious to a lot of you who know this stuff, but, in Phase 1, we're looking at short and long ASCII domain names. Phase 2 is Unicode, IDNs, and the A-labels that are associated with this. Phase 3 is EAI. That is a Phase 3 for us because that is probably the most complex for us. So this is kind of in order of complexity.

As I mentioned, we have push to our major vendors. Our CIO, Ashwin, is involved and engaged and he is helping us work with our major vendors to make sure that they are aligned. So he is speaking with his counterpart at various major vendors [inaudible] to make sure they know what we're expecting in working with us. Secondarily, I am focused on our more minor vendors and working through some training and communications with them. So there is, at all levels within ICANN, a real push to make sure we get those vendors ready to go.

Next slide. There are a couple slides coming up here just about where we're at at a very high level. Now, this slide represents how our custom in-house systems that are public-facing. On the left there, you can see just a general pie chart about our business-directed services. About 60% of them do not have a need for UA readiness just because they don't use e-mail. They're internal-based-type system. So they're supporting-type systems. There's no need for them to be UA-ready.

Of the ones that are remaining, we essentially split about half and half as far as our custom in-house services and our off-the-shelf services. This slide addresses custom in-house. So, sorry, those arrows are a little bit misrepresented there. Both arrows should be coming from the custom in-house section.

As I mentioned earlier, Phase 1, Phase 2, and Phase 3 are addressed. Phase 1 is the gTLDs (short and long) and Phase 2 is Unicode and IDNs.

You can see—we've made this claim previously—that, with Phase 1, we are done. All of our systems that need to be UA-ready are UA-ready. For Phase 2, we have about two-thirds of our systems that are ready to go and supporting IDN and Unicode. We have some … This again is specially about URLs and not the EAI. That's out of Phase 2. That's Phase 3. About 28% don't use that. They may only use, say, a mailing list or something like that. So they're actually not applicable for Phase 2.

And there's a very small chunk that are in process, and they're actually, even though they're custom in-house systems, reliant on the external vendor for authentication. Specifically, we have some single sign-on, and there's on particular vendor that we're working with there that is not UA-ready yet. So that affects this graph, although the systems themselves are UA-ready. So we are working diligently with that vendor to make sure they are ready to go and they are working on that. We're working with them to make sure that they get up to speed very quickly.

Next slide, please. This is our off-the-shelf. This is our external vendor systems that we use. Of all the services that are to be considered for UA, there's a split there. You can see a third to two-thirds. A third are only affected by Phase 3, which is EAI, and, for about two-thirds, the Phase 1/Phase 2 components.

You can see generally there that we are making progress. For Phase 1, we're in review. For some of those, the [end] review there does not reflect that they are not UA-ready. It reflects that are working with them. We are working through some test examples to verify exactly how many still need to do some work. So we have verified that about two-thirds are ready to go, and the rest are in review. For Phase 2 (the Unicode/IDN piece), you can see it's about 60/40 there. Again, really a big focus for us are the external vendors and making sure they are ready to go and understand what it means. So that is really where a big push for us is happening. As you can tell from the previous slide, internally we're pretty comfortable with being covered. It's really the vendors that we need to work [with].

Next slide, please. Last here is what I categorized before as Phase 3, which is EAI support internally for ICANN. We are definitely making progress here. Our clients are ready to go. Across the Org, we support UA with our clients. We have a couple of things. Obviously, as we had talked about previously, there's a lot of components to supporting this: our exchange server and our routers and gateways. We are relying obviously again on vendors here, but we do have commitments and we expect those to be ready to go by Q4 FY'20, which is about the June timeframe. Q4 is April to June this year. So those we expect by June to be ready to go.

The last one for us, which has been a challenge, is our security/spam filter vendor. They have not supported as quickly as we would have liked. We are working them actively. Ashwin has been involved with them and had lots of call with them to work them and move them forward. They actually now have assigned an engineer, and that engineer is actively working on their solution to [inaudible] for us. So they are our longest pole in the tent, if you like. They are the ones that we're waiting for.

However, having said that—we're still waiting for commitment from them on a timeframe—with the exchange server and [routed] gateway being in place by about June, we will be able set up some internal testing. We'll effectively bypass the spam/security filter for our internal testing. So we can start our EAI testing in earnest for all of our systems in around the July timeframe. So we will be doing that. I think, at the last ICANN meeting, somebody had asked the question, "How do you know, one Day 1, when you're finally ready with the security/spam filter vendor, that everything will be compliant?" So that will be the big effort we do starting in July to make sure we are ready to go as soon as that happens.

Another big platform for us, for those of you who know, is Salesforce. We have a lot of our partners that are on the Salesforce platform for what we do with registries and registrars. Salesforce has announced that they will be EAI-compliant in the Summer '20 release. So we will obviously follow that.

One more slide, if you switch to that. Next slide—thanks. Really, this slide is just to announce that we are working on an update to our case study. There is an ICANN case study that's on the UASG website today. It's a couple years old. We have updated that. That is in production, effectively, at this point. That will be published shortly after the meeting. So that is reviewed and is in process. That will be made available to the UASG and will be on the website.

That's it from me. I think we're taking questions at the end. Is that right?

SARMAD HUSSAN: Yes. Thank you, Dean. We actually had a hand up, but I don't see that anymore. If you'd like to—okay. Sivasubramanian, would you like to comment?

SIVA MUTHUSAMY: Yes. Just a question. I heard the panelist mention that tax systems and certain government systems do not have a need for UA readiness. Could you elaborate on that, please?

DEAN EDWARDS: Is that a question for me? For Dean?

SIVA MUTHUSAMY: Yes, it's a question to you.

DEAN EDWARDS: I did not mention government systems. I'm not sure what your question refers to. I apologize. Maybe I didn't communicate clearly. There was a slide where I was talking about some of our public-facing systems not being applicable. That is because they do not process e-mail addresses. They do not use URLs in the function that they provide. So there is a number of our systems that just do not need to do be UA-ready because they do not use the components that are required. So there was nothing in there about government systems, but that perhaps is what you caught: that statement.

If that's not it, maybe you could clarify.

SIVA MUTHUSAMY: No, that's it. But I can't imagine any taxes [or] any kind of form without an e-mail address component because every applicant or every person who fills in the form or files a return is expected to provide an e-mail address. Besides, it communicates by e-mail, which could be an IDN. It could be something with a [long] domain name. So I'm not sure why you concluded that some of the systems don't have a need for UA readiness.

DEAN EDWARDS: Well, we have some systems that are informational. We have websites that don't require an e-mail sign-up. Things like that. So there are several ICANN sites and services that do not need to have e-mail to be able to access them.

SARMAD HUSSAIN: Okay—

PITINAN KOOARMONPATANA: We have two more questions in the chat.

SARMAD HUSSAIN: Pitinan, let's take some questions from the chat.

PITINAN KOOARMONPATANA: Sure. I'll read the two questions all together, the first for Marc Blanchet. "I'm surprised routers are really involved in this. Can you explain what are the issues with the routers and UA readiness?" The next question is from Ajay Data. "Dean, please update [on when the mailing list being EAI-ready is likely to be." End of question.

DEAN EDWARDS: On our IT system, what we're talking about here is, internally, within ICANN, how we process e-mails. This is a pretty typical setup where we use an Outlook client for our end users. We have an exchange server/routers/gateway that route the messages internally within our network and then the security/spam filter. So those are the components of our mail system at ICANN. Pretty much all of those are external vendor-supported systems. So there's a requirement there for all of them to be able to handle international e-mail addresses. This is not an unusual issue, but our vendors today do not support international e-mail addresses, or they are in the process of testing or they're working on releases that are upcoming.

We also have to, as you can imagine, schedule any updates to our system. So, if we have to do software updates, updates to our infrastructure, that is something that has to be scheduled and worked through and we have to test it before we deploy widely.

So this is a timing thing just around what our planning is for our IT team at the moment and updating our infrastructure to support international e-mail addresses. Hopefully that answers that questions.

Then, on mailing lists, that's a point we've been talking internally on quite a bit. We currently use a vendor or we use open source that is not yet ready to do that. There is some talk about what we can potentially do if there's a way for us to migrate to a different mailing list service. At the moment, that's in review. We don't have any commitment yet. We're certainly working through that and trying to understand what our options are. There aren't a ton of mailing-list-type vendors that we can use today that do support it. So that is definitely one we're aware this group is looking for and is something that would really be a good thing to have. So we are actively working through that, trying to understand what our options are and if there is anything we can really do about that.

I apologize if I don't have a specific update, but that is something we are definitely looking at to try and resolve, if we can.

SARMAD HUSSAIN: Thank you, Dean. We are running short on time. We should move on. I saw a comment from Ram. Ram, you want to quickly comment?

Not hearing from Ram, let's continue moving on. There was a hand raised by Mark, but, Mark, we can move on to an update on the UA-EAI Working Group. You can also make your comment, if you want to do that now. Over to you, Mark.

MARK SVANCAREK: Hi. This is Mark Svancarek. I'm the Chair of the UA E-mail Address Internationalization Working Group. I did have a comment from the previous thing about governments' requirements for UA and EAI, but I think the conversation has moved on from there. I would just say that we're not seeing a lot of government demand right now. I do think that many governments would benefit from it. But, specifically for EAI, I'm not seeing a lot of demand right now. Surprising, but …

Anyway, next slide. What we're planning to talk about is the progress of our group, future work that we're looking at, and then a discussion on our best-practices document.

Next slide. We're a relatively new working group insofar as we got a later start than some of the other working groups. Here are the outcomes that we decided that we would like to achieve within this fiscal year.

1, 2, and 3 are well in hand. The first one: review the levels and phases of EAI support already developed and refine these as necessary. This specifically goes to the concepts of Phase 1 and Phase 2, namely, can you send to or receive from an EAI-compatible e-mail system? That's Phase 1. And can you actually provision such mailboxes yourself, which can then subsequently send to and receive from? That would be Phase 2. So those levels of phases remain in place. We've just extended them beyond an entire mailbox system to talk about the various components and various tools and libraries and stuff like that.

The second one is developing the inventory of proprietary and open-source tools and applications for EAI. That has been generated as a statement of work, which is currently flowing through the ICANN system. So you should see that on the list relatively soon, and, related to that, the inventory of e-mail service providers along with their contacts. So that's actually a two-phase thing. First, develop the inventory of e-mail services providers. Secondly, see which ones have contacts available and find a way to bring them into our community or not. Again, the statement of work is finalized and moving through this system. So you should see it on the list.

Next, developing the technical requirement for implementing EAI and the baseline metrics for measuring EAI. So you should look at what's happening in the Measurement Group. There is an EAI document. I think it's UASG 21A and UASG 21B. You should take a look at those for additional context.

Number 5 is to undertake a review of the technology and services to determine the degree of their support. That's in conjunction with the Measurement Group.

So, if you break this down into systems—MUAs, MTAs, various third-party support and utilities—it's a pretty long list of things. It's a matrix. So you're looking at all the different service providers and then all the different components of them, as well as tools, suppliers, etc.

So that statement of work was announced. It has made it all the way through the system. It was announced on the list, and we do have one person who has submitted a quote for that. As you can imagine, there's a huge amount of testing that's involved here—many, many hours—so the statement of work is comprehensive. Also, it's a fairly big amount of money. So we're looking at ways to prioritize that—maybe do it in two phases. But we do have one vendor who has put forward a bid.

Finally: review, update, and develop technical documentation and make the translations available. We do have an EAI deployment tutorial, which was mostly developed by org. That's been posted on the list. We've been soliciting feedback on that for a couple of weeks now. I think it's probably ready to move to the next step and get some translations done. If that's not a good assessment, Sarmad or [inaudible] can clarify that.

Seeing no concerns on that, next slide, please. #7 is related to the best practices and standards for the security and usability for the e-mail providers. This is an ongoing work. How do you deal with homographs, script mixing, variant characters, variant strings, etc.? I've got a link here if you want to look at where we're at right now.

It's interesting. We have a lot of work going on in the domain name portion of universal acceptance. Our focus in our working group is really on the local parts. There is an overlap, of course, with what happens in the domain names part, but we don't want to be redundant and redo the work. So we're focusing on the local parts.

It's interesting—at least I think it's interesting—that a lot of our vocabulary is focused on the domain name parts. When we talk about variants, variants is a concept that's well-defined within the top level and the label generation rules. When you discuss a concept similar to that in the local part, it's not clear that people will understand that it's well-defined in one context and not so well-defined in the other context, even though we're mostly trying to solve the same problem—not entirely.

I think you've seen on the list already a comment from Andrew Sullivan that the local parts have very few restrictions at all in what you can put in there. So the concept of variants is not entirely subject to the label generation rules, which are about very much constraining what sort of strings exist there. So we can talk about them in concept. We can give best practices for security and avoiding confusability, but they're not entirely the same.

Likewise, we advise people: don't start local parts with xn--. It's not actually restricted. Of course, you can start a local part with xn--. In fact, I have one on [Hotmail]. But we're saying it could be confusing to software systems, it could be confusing to users, and therefore we advise against it. But what do you call that exactly? It's not Punycode. Punycode is defined for the domain name part. So we struggle with this sort of guidance in our documents. I think we're making some progress. It does mean that we use more words than you think we might have to, just because we want it to be really clear that we're focusing on the local parts instead of the domain name parts.

There was a comment earlier in one of the previous sections about the ASCII@IDN.IDN. Now, if you look at our technical documents, I think I've got a link here. Let me pull up the link. So this is UASG 12. On the downgrading section, there is a comment that there is an uncommon case where the local part is explicitly defined as ASCII, and the domain name part is explicitly defined as UTF-8. I don't actually know what that looks like in actual practice. I'm not sure I've ever seen one. But John Levine included it in the document for completeness. I don't think we've ever seen one of those.

So that's one of those cases where we're saying there could be a conceptual ASCII at IDN.IDN, but most of the time, the local part will be normalized as UTF-8. Then you have characters that are within the ASCII principal character set, but they're actually Unicode. So there could be a conceptual confusion about what do you call that exactly. Do you call that ASCII in the local part? We try to say that that's Unicode in the local part.

So that's a long description. Just look at the link. You'll see where we are. We're not done yet, but the draft is coming along.

#8: Prioritizing and outreach to the tool and services contacts. I think I mentioned that already. As I stated already, we have to have the inventory in place before we can do the second part of the work. So we'd like to know who the contracts are. We'd like to bring them into the community. Our privacy policy prevents us from getting a bunch of contacts and posting them on the list. So we need to get them into the community in order to put their support structures in place and allow the community to talk to them. For instance, if you want to talk to Microsoft about support for EAI or IDN, you have to go through the public support network. We're not just sending it to me anymore.

Finally: Developing the training materials for the deployment for e-mail administrators—their tools, their applications developers, and things like that. That will be ongoing as well.

I see some questions in the chat and some statements. Let's get to those in the end, if that's okay.

Next slide. All right, the last ones. These are the things that are not even underway. I hope we will get to these during the fiscal year, but you know how things can sometimes take longer than you expect. So: Developing and undertaking the training dissemination. That'll be started after the materials are finalized. Developing the communication materials to acknowledge the support of EAI. There is a questionnaire that the Comms Group put out. I hope you will respond to that. That's then first part of this developing of communication materials.

Finally, as we find software and service providers who are EAI clients, we would like to recognize and celebrate them. We should celebrate our victories. Hopefully this encourages other people to participate. But, even if it doesn't, we would like to recognize the software and service providers who are UA-compliant who do participate in our working group, etc.

Finally, as I said a moment ago about how long things take, we are usually meeting on a weekly basis rather than a biweekly basis so that we can catch up to the other working groups. I think this has been very helpful to us.

Next slide, please. In summary, our future plan has these three sections. One is building the capacity, which means to develop and disseminate the training material. So how do you add support? How do you deploy it? We're looking at webinars, training courses, face-to-face trainings, and other effective mechanisms. Some of these have already been started by the ambassadors. You'll notice this in the ambassadors and local initiatives section. But there's much more to be done, particularly for EAI.

Then we'd like to communicate our success in collaboration with the Communications Group and, as I said before, recognize and celebrate our alliances. So, if you are compliant, we'd like to make sure everyone knows about it.

I think this is our last slide—oh, no. The best practice for mailbox accounts. I've already touched upon this, but let's go ahead. Next slide. Here you can see what I was mentioning. What do we want to call these things? You could call the thing to the left of the @ sign the mailbox name, the account name, the user name, the UTF-8 part (that's not a good one), or the local part. We decided we're calling it the mailbox account name. It took us a while to decide which one of the things we wanted to use. We wanted to be consistent and we wanted a name that would be least confusing and most familiar to the greatest number of e-mail admins possible. So we recognize that these other strings are available. We call that out, but we're focusing on mailbox account name.

As I mentioned before, script and relevant terms. When we're talking about things in the old ASCII—principal character set, for instance—when we're talking about other characters, and the character sets within scripts, we refer to the Unicode standard.

Finally, the whole concept of validity and variants. So should we suggest to an admin that there should be a length requirement? In the end, we decided that, no, that would be overreaching our recommendations. So all we said was, if you have a length requirement—mailbox account can only be eight characters or something like that, if that's your internal rule—you probably should apply it to types of e-mail, not just your old ASCII ones but your EAI ones as well. But that's just a suggestion, not necessarily a best practice. But consistency is always good.

Regarding variants, how do you suggest managing variants? Because, on some scripts, it could be a huge number of tests. So we want you to consider creating policies for that while supporting as many as makes sense for your local script and your local culture.

Script mixing should be prohibited. We had to think about, what if you're using the Japanese character set, which includes the Roman characters, which are essentially the ASCII-printable character set? But, luckily, Unicode defines how that works. So, as long as you're referring to Unicode, you'll understand how to manage those things. That's what the second bullet is saying.

Avoid security issues which are potentially caused by invalid labels and their variants. How do you determine what is invalid? The mailbox account rules are very, very liberal. There are almost no restrictions on it at all. Obviously, you can't have a mailbox name that includes [inaudible] character, but there's not a lot of other ones like that. So we've been trying to come up with definitions of what you should not include in your mailbox system.

If you have any suggestions for these, please look at the document and submit your suggestions.

Now, the second bullet. We talked about this quite a lot. Are the label generation rules applicable to a mailbox name? Our suggestion is that mailbox names are much less restricted than label generation rules. So, if you elect to use the label generation rules when deciding which mailbox account names are valid or invalid, you really can't go wrong. That will help you manage the confusability and security issues, but it might be too restrictive for your local culture. You may find that people really want names that go beyond what the label generation rules allow. So that's our suggestion: look at those label generation rules. If you want to apply them, you should be in a very good, safe situation. You may find that they're too restrictive to you. So your mileage may vary, as the phrase goes.

Now, the third bullet is related to this root zone label generation rules.

Next slide, please. Here are some of the examples where label generation rules are too restrictive to many people: disallowing digits at the beginning of a label, use of digits in general, and, of course, if you're using a script that doesn't have a label generation rule defined yet. For that, we suggest you look at RFC 6912.

Next slide.

SARMAD HUSSAIN: We are a bit short on time, so—

MARK SVANCAREK: Yeah. I think we can skip this. I've already talked about it at length. And you can skip that again. That's just the one that says, depending on your script and how many variants you're supporting, you could wind up with a huge test matrix. So you should be considering that upfront.

Again, skip ahead. This is the Punycode one. Please skip ahead.

Here's the next set of discussions that we'll be having in our document. How do you manage the special characters? Should we talk about normalizing the normalized form in this document? And when is it a good idea to associate EAI mailbox names to ASCII mailbox names? I think you'll see that a number of vendors—XGen, for instance—allow you to do this. I think many systems allow you to associate multiple e-mail account strings to a single mailbox. So this is a similar concept.

Next slide. All right. How much time do we have for Q&A, Sarmad?

SARMAD HUSSAIN: I think it may be good to go through the next set of slides and then open for Q&A for everybody. Otherwise, we may not actually get time to go through the next set of slides, if that works.

MARK SVANCAREK: I think that probably works. I'm sorry to everyone in the chat who has been asking questions. Hopefully, we'll get back to you in the end. Thank you for this opportunity to tell you about the EAI Working Group.

Back to you, Sarmad.

SARMAD HUSSAIN: Thank you. I'll invite Mark to maybe quickly take us through the EAI training being done, and then we'll get to the questions and answers session.

MARC BLANCHET: Thanks, Sarmad. We'll be quick. The next slides are about EAI training. The goal is create training material targeted to system administrators. This is key for this training. This is not for end users but for system administrators to enable EAI. 77 slides. Configuration examples, fully functional demo setups, and some deployment considerations.

Next slide. Topics that are discussed in the tutorial are, for example, configuration examples of open source software that could be installed by sysadmins. We tried to introduce key concepts and considerations to be relevant for the sysadmins if they are not using the software shown in the examples. Obviously, there's thousands of different e-mail systems. But still, the tutorial is useful for the other software that are not shown in the presentation.

It also shows the details of the mail headers that are changed for EAI, which is useful for them to debug issues because you will be debugging stuff. And there's some considerations related to deployment.

Next slide. Table of contents. The same kind of small set of key concepts that are useful for the sysadmins. It discussed EAI changes to protocols—SMTP and IMAP and POP—for supporting EAI, some development considerations, and a demo setup with Postfix and Courier, which are two open source mail, and Mailmate and Gmail. So you get all those components talking to each other.

Next slide. Going fast. There are a few things that are in the tutorial. A key point/consideration that people need to know is that all servers in the mail delivery path must support EAI for the e-mail to reach its destination. This is small diagram of that all of them should be supported. It discusses also the fact that you cannot [inaudible] infer the mail delivery path because, depending on availability of servers, they may use a different path every time.

Case folding in UTF-8. EAI is not typically available by default in software. Therefore, people may be used to case folding in ASCII, but that's not the case by default in UTF-8 EAI.

If an IDN is using an e-mail address, servers must be configured to support both encodings. And EAI is often considered as spam by the spam filtering engines. I think our friend from ICANN actually discussed that.

Next slide. Going fast. And that's it. So EAI is essentially done. The EAI tutorial for sysadmins is done. We should producing the final version there soon.

SARMAD HUSSAIN: Thank you, Marc. Let me hand it back to Dr. Ajay Data. We have five minutes left based on his direction. We'll take some comments and questions.

AJAY DATA: Thanks, Marc. Thank you, everyone, for joining in such a large number. A very interesting, active chatroom, I can see.

I want to take this opportunity to [inaudible] Maarten Botterman, the ICANN Chair I can see on the list, if you'd like to take a minute to address everyone and speak. I'm very happy to see you here and the support you have for UA. Maarten?

MAARTEN BOTTERMAN: Thank you, Ajay. I'm very pleased to see this. As you know, I've been following universal acceptance for a long time—actually since 2006, the first time that I talked about it. It is great to see how the progress is. I also love the commitments that have been expressed here. I'm participating [with lots] of them. I think the participation in the room is maybe even bigger than normally would have been in Cancun, which is good.

Not taking too much of your time, because questions between each other are more important than my remarks, but, from a personal perspective, there's two thoughts that are came up—just thoughts—for you to do with what you want or not. One is that I think the testing on UA readiness is all amazing and may benefit from also being promoted on other testing platforms that are out there for Internet tech testing. So that's one thought. The other one is I still hope to see more IDN content because that will drive demands. It'd be great to see in the meanwhile how to deliver when the demand is there. It's moving up.

So thanks. Back to you, Ajay.

AJAY DATA: Thank you very much, Maarten. I really appreciate the leadership you're providing and allowing us to run these [inaudible] meetings like that. Thank you very much.

Sarmad, back to you, please.

SARMAD HUSSAIN: Pitinan, please go ahead.

PITINAN KOOARMONPATANA: We have a few comments. A comment from Bill Jouris on the various definitions. Comment: “[No] variant is not well-defined in the domain names area. In fact, the definition has changed continually over the past three years.”

I’ll just go ahead. Next comment, from Abdulmonem Galila: “EAI development training [inaudible] already done and tested well. Completely open source packages. I did already a hands-on EAI development workshop here in Egypt recently. We’ll talk more about this during Wednesday’s sessions.” End of comment.

The last comment, from [Jim Delahunt]: “I think we had as many as 92 online participants in this session. Yay!” All right.

Then I’ll move on to the questions. We have three outstanding questions. First is from Ram Mohan on the EAI topic: “Can we presume that we are not promoting interesting Unicode-valid local parts, like emojis, as part of EAI?”

I will also pause the next two questions in the chat so the speaker can see and prepare the answer. Thank you.

SARMAD HUSSAIN: Mark Svancarek, would you like to take that question?

MARK SVANCAREK: I’m sorry. Which question was that? I didn’t think it was for me. Sorry. could you say it again?

PITINAN KOOARMONPATANA: Sorry. It's from Ram Mohan: "Can we presume that we are not "promoting" interesting Unicode-valid local parts, like emojis, as part of EAI?"

MARK SVANCAREK: Sorry. That was for me, obviously. No, we are not promoting that. I forget what exactly our guidance is, but we're not. Thanks.

PITINAN KOOARMONPATANA: Thank you. The next one, from [Yunia], I guess is for Marc Blanchet: "Question as to your EAI readiness conformance. Does UA conformant, Regex, need to be matched variants? For example, [if Regex is equivalent in] NFC/ NFCK capability." End of question.

Also, some were in the chat comment. "Background of my question. No boss operators don't know what the variants are, and they probably need automatic variants matching functionality by NDS software. Here variants are not only the NFC/NFKC compatibilities but also for SC/TC confusability and so on." End of comment.

MARK SVANCAREK: I do generally agree with that. I think I touched upon that a little bit. It is a challenge to come up with the recommendations, particularly with the absence of the tooling to help you do the management. But I would say that this is a larger problem not just related to mailboxes but a situation that you see with IDN variants as well. The tools are not mature enough to allow you to really generate all your variants and decide which ones you want, at least not that I've seen in the last year. Thanks.

PITINAN KOOARMONPATANA: Marc Blanchet, do you also want to come in for that?

MARC BLANCHET: I think [inaudible] really is something which is not a simple fix. Regex for variants—yeah. I have no good answer at the moment, apart from—yeah. A difficult problem.

PITINAN KOOARMONPATANA: Thank you. I'll move to the last one. This is for Dean Edwards, I guess, from Sayvo Morai. Question: "ICANN processes of dealing with operational issues is going to be published? For example, the case of security/spam filters. I think this is important for the community to learn how to deal with it, not only in the software development processes but on the day-to-day maintenance, too." End of question.

DEAN EDWARDS: I think that's a very interesting comment. I think that could potentially be valuable. I agree. As I had mentioned before, we are reliant on some vendors there, so we're not doing development specifically. But I think it's an interesting point to maybe put something together just around the issues and what were the challenges for us. I think Marc also mentioned, when he was talking about EAI support, that the spam filter is one example, but there are obviously a lot of components just to get the flow down and everything supported end to end.

So let me take that away. I'll talk to the team. I'll talk to Sarmad and Pitinan. We'll work through that. I think that's a good suggestion, so thank you for that.

PITINAN KOOARMONPATANA: Thank you. We have one question, but I guess it was well-discussed on the chat already from [Morai] about the UA readiness deployment. "We need policies and other things besides technical."

So I guess we are over the time. If there's anything we haven't covered, please feel free to contact us.

I'll pass it to you, Sarmad, to close.

SARMAD HUSSAIN: Dr. Data, would you like to come in and have the last comment?

AJAY DATA: I think we had a wonderful meeting. Thank you, everyone, for joining. Actually, I wanted to say something related to the comment which Siva and one more person made about variants and IDNs having entire Unicode. Obviously, we all know probably that IDNs are limited to the [inaudible], apart from some exceptions ([NCJK] and some other scripts) where they can exchange some specific characters. Otherwise, IDNs [inaudible] different rules for each step, and the IDN can be in the data script alone. So the entire Unicode label [or code points] are not likely to be in IDNs, at least for now. Mailbox is a different story, for sure.

Related to variants, I think people are following the rules, and you are right that software developers who are building the solution for mailbox names and accepting an IDN are going to implement the rules for how the mailbox matching and variants are going to be dealt with. So it is not going to an administrative task to worry about that, and they don't have any worry today. So this is going to be dealt with in the mailbox [inaudible] situation. Companies are doing it, and companies are solving that problem very effectively—those who haven't gone through it.

Thank you very much for joining. Thank you very much for attending this call. We have two more sessions coming up in this week. See the schedule. I'm looking forward to everyone to join these sessions. See you again soon. Thank you very much. Enjoy.

SARMAD HUSSAIN: Thank you all for joining. We'll close this session.

PITINAN KOOARMONPATANA: Thank you, all. Bye-bye.

[END OF TRANSCRIPTION]